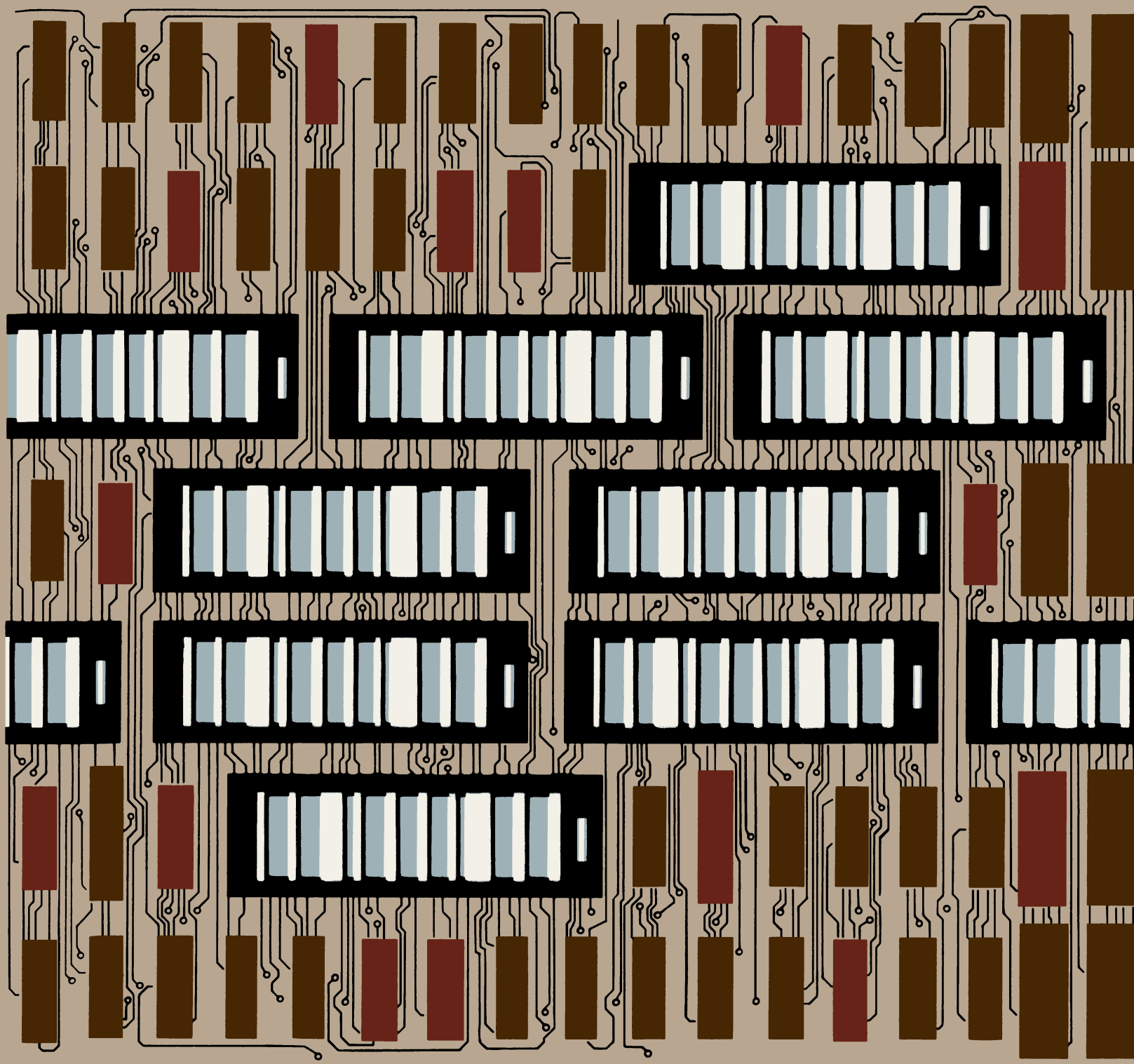
VAX-11/750
UNIBUS Interface Technical Description



digital

BLANK

EK-UI750-TD-001

VAX-11/750 UNIBUS Interface Technical Description

digital equipment corporation ● maynard, massachusetts

First edition, August 1980

The material in this manual is for informational purposes and is subject to change without notice.

Digital Equipment Corporation assumes no responsibility for any errors which may appear in this manual.

Printed in U.S.A.

This document was set on DIGITAL's DECset-8000 computerized typesetting system.

CONTENTS

FIGURES

FIGURES (Cont)

TABLES

CHAPTER 1
INTRODUCTION

1.1 SCOPE

This manual provides a technical description of the UNIBUS Interface (UBI) subsystem of the VAX-11/750 processor. The three chapters provide general, functional, and logical descriptions of the UBI. Brief descriptions of the UNIBUS and CPU/Memory Interconnect (CMI) are included. Prior training or experience with VAX architecture is assumed. The manual is intended to be a resource for branch and support level courses, field service and manufacturing, and as a general reference document. Table 1-1 lists related hardware documentation.

Table 1-1 Related Hardware Manuals

Title	Document No.
VAX-11 KA750 Central Processor Technical Description	EK-KA750-TD*
VAX-11 MS750 Memory System Technical Description	EK-MS750-TD*
PDP-11 Peripherals Handbook	EB05961†
VAX-11/750 Hardware Handbook	EB17281†

* Available on microfiche.
† Available on hard copy.

Hard copy documents can be ordered from:

Digital Equipment Corporation
444 Whitney Street
Northboro, MA 01532
Attn: Communications Services (NR2/M15)
Customer Services Section

For information concerning microfiche libraries, contact:

Digital Equipment Corporation
Micropublishing Group
12 Crosby Drive
Bedford, MA 01730

Manual Organization
Chapter 1 is an introduction to the UNIBUS and the CMI, and all logic resident on the UBI module.

The UNIBUS Interface (UBI) section of the module is presented in Chapter 2, Functional Description, with descriptions and explanations of major circuit functions at the block diagram level:

- UNIBUS Data Paths (UDP)
- Address Map (MAP)
- UNIBUS Control (UCN)
- UBI Control Store

Chapter 3, Detailed Logic Description, describes UBI microsequencer operations within the VAX-11/750 system:

- CMI initiated transactions
- UNIBUS initiated transactions
- UNIBUS interrupt/write vector transactions

Chapter 3 also includes a brief description of non-UNIBUS logic that supports processor functions. Tables of backplane jumper selections are included. Functional descriptions of the logic listed below are provided in the VAX-11 KA750 Central Processor Technical Description (Table 1-1):

- Interrupts (INT)
- Console Interfaces (CON) to console terminal and TU58
- Time of Year (TOY) clock
- System hardware revision level field of the System Identification (SID) longword

1.2 UBI SUMMARY
Figure 1-1 is a diagram of the subsystems that make up the basic VAX 11/750 system. The UBI module contains the UNIBUS interface to the CMI lines and the logic for some system functions. The UBI module is an integral part of the CPU. Figure 1-2 illustrates the UNIBUS interface and other significant logic on the UBI module in the context of the CPU. Further system information is available in the VAX-11 KA750 Central Processor Technical Description, EK-KA750-TD, and other related manuals listed in Table 1-1.

1.2.1 UBI Functions
The UNIBUS Interface (UBI) serves three purposes.

1. UBI allows the processor to access registers on the UNIBUS.

2. UBI allows devices on the UNIBUS to perform DMA transfers to main memory.

3. UBI allows UNIBUS devices to interrupt the processor.

(Further information regarding the UBI and its facilities is available in Chapter 8 of the VAX-11/750 Hardware Handbook, EB17281.)

Several characteristics of VAX architecture and the VAX-11/750 main memory system require more than a straight-through connection from the UNIBUS to the CMI. These characteristics are discussed in the following paragraphs.

First, addresses that are contiguous in virtual address space may be discontiguous in the physical address space on 512 byte boundaries. Since all UNIBUS nonprocessor request (NPR) devices broadcast sequential addresses, a means is provided to map these addresses into disjoint 512 byte blocks.

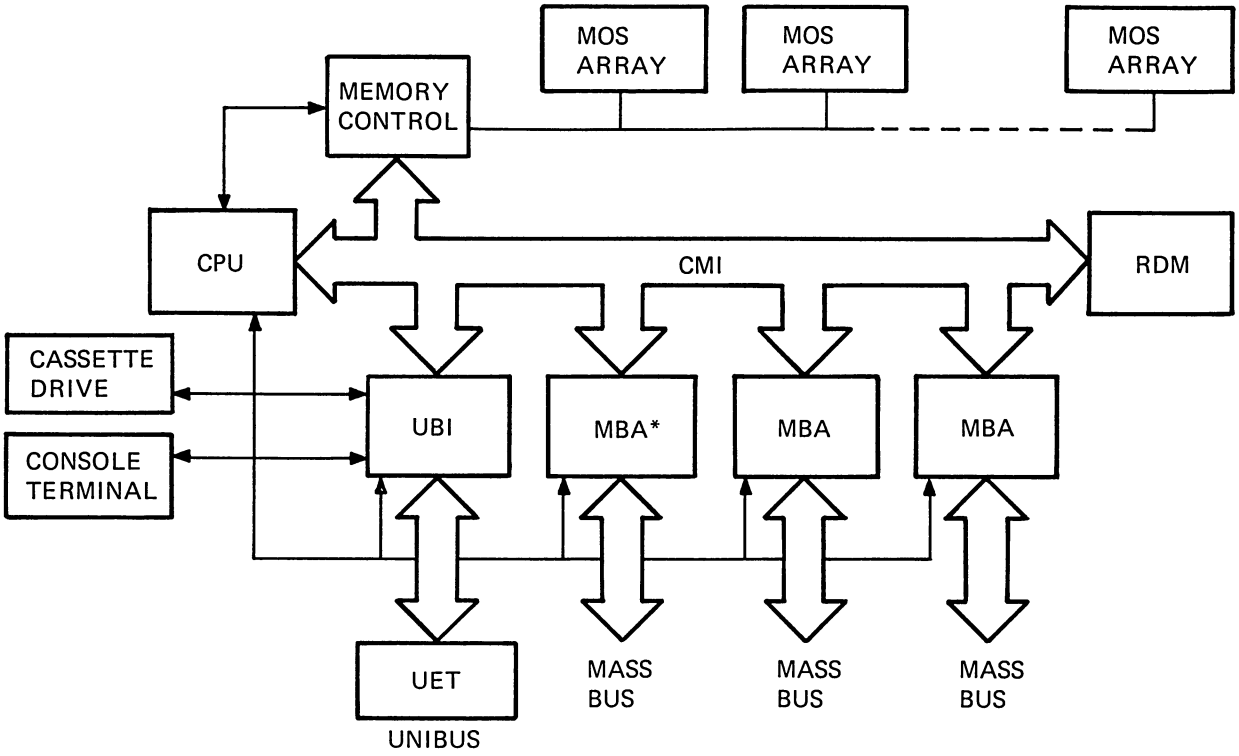


Figure 1-1 Basic VAX-11/750 System Diagram

Second, VAX architecture imposes no restrictions on the alignment of data in memory. UNIBUS NPR devices, however, only transfer word data on even addresses. A UBI mechanism allows the transfer to be shifted by one byte to accommodate requests for I/O buffers on odd byte addresses.

Third, the CMI has 24 byte-address bits; the UNIBUS has 18. The UBI provides the facility that allows a UNIBUS device access to all CMI address space.

Finally, the CMI is four bytes wide; the UNIBUS is two bytes wide. Utilization of both CMI and UNIBUS is improved by compressing two sequential UNIBUS transfers into a single CMI transfer.

UNIBUS NPR transfers take place on one of three buffered data paths (BDPs) or on the direct data path (DDP). The main advantage of a BDP over the DDP is that fewer CMI cycles must take place to transfer data between the UNIBUS and main memory. Each BDP consists of a data buffer register of four bytes, sixteen bits of address storage, five flag bits, and the necessary control gating and logic. A BDP buffer register effectively acts as a small cache.

The selection of a data path is dependent on the value specified in the address map. The DDP is selected when the data path select bits in the address map specify 0. Data is gated directly between the UNIBUS and the CMI and no data is stored in the UBI. Further, SSYN is not issued by the UBI until the corresponding CMI transaction is completed. One of the buffered data paths is selected when the data path select bits specify 1, 2, or 3. Only one CMI transfer is needed for every two UNIBUS word

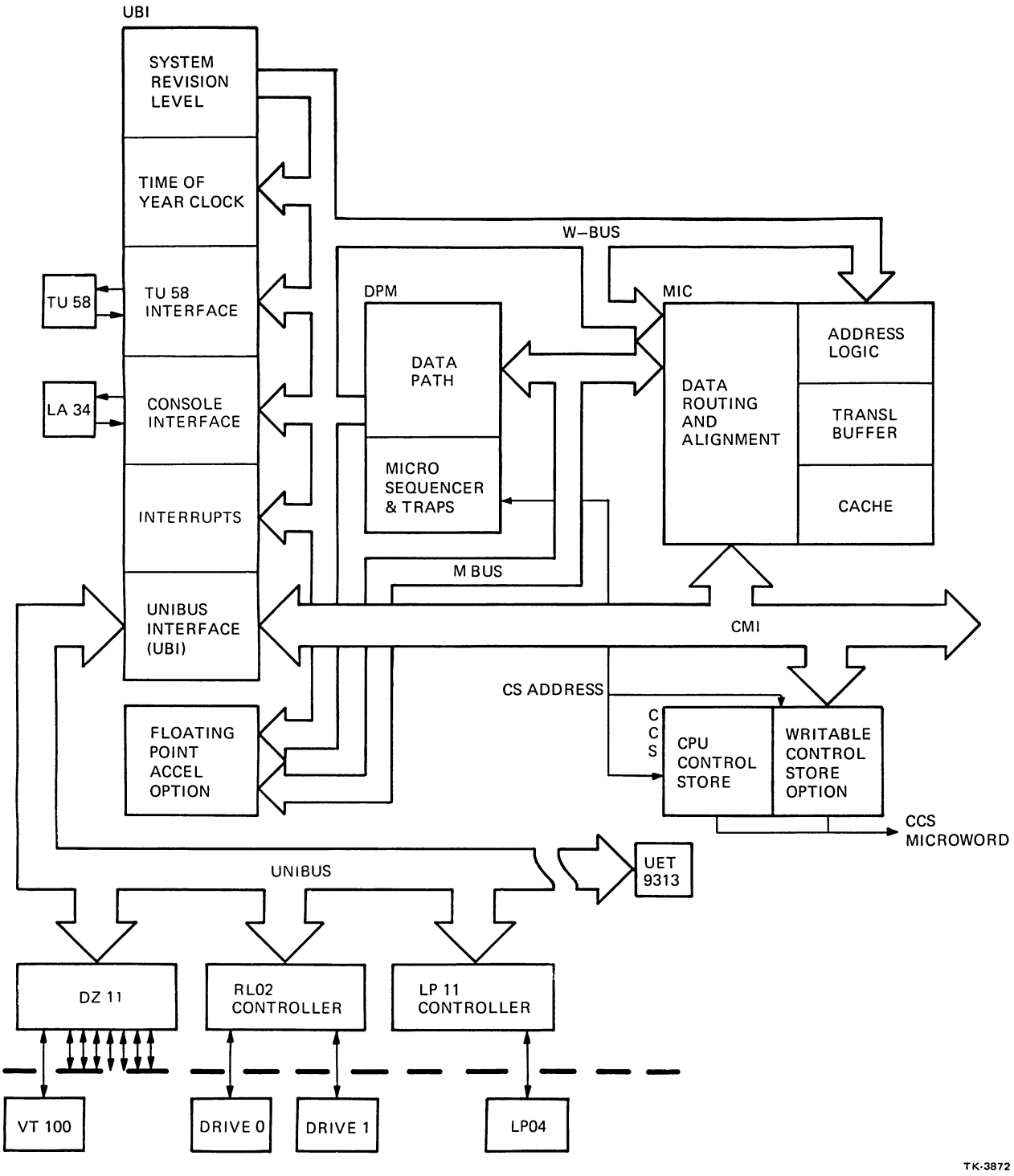


Figure 1-2 CPU Block Diagram

transfers. Also, data can be loaded one or two bytes at a time from the UNIBUS, but data is always transferred four bytes at a time on the CMI.

1.3 THE UNIBUS

Figure 1-3 and Table 1-2 provide descriptions of the 56 UNIBUS (UB) signals. These signals are divided into three functional groups: data transfer, priority arbitration, and initialization. The last peripheral device on the UNIBUS contains an M9313 UNIBUS exerciser/terminator module (UET).

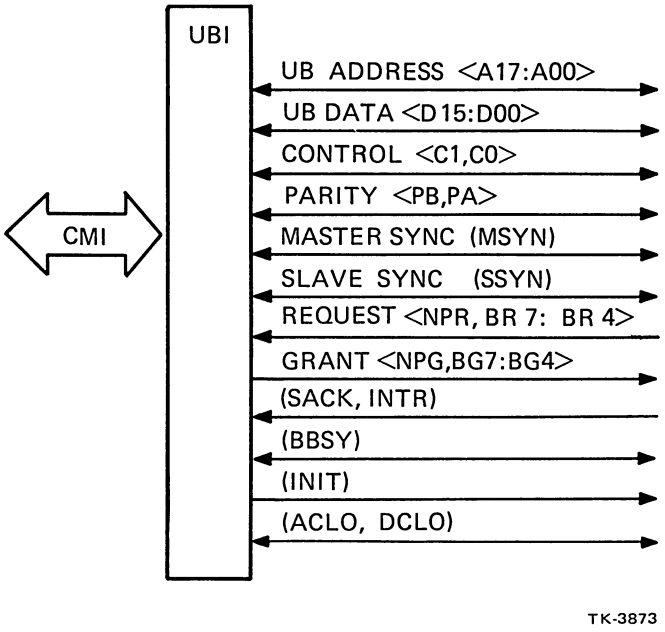


Figure 1-3 UNIBUS Signals

Table 1-2 UNIBUS Signal Description

Signal Line	Description
Data Transfer Group	
Address Lines ⟨A17:A00⟩	Address lines are enabled by the master device to select the slave (actually a unique memory or device register address). ⟨A17:A01⟩ addresses a 16-bit word; ⟨A00⟩ specifies a byte within the word.
Data Lines ⟨D15:D00⟩	Data lines transfer data information between master and slave.
Control ⟨C1:C0⟩	Control lines are coded by the master device to control the slave in one of four data transfer operations. Transfer direction is designated with respect to the master device.

Table 1-2 UNIBUS Signal Description (Cont)

Signal Line	Description
Data Transfer Group (Cont)	
C1 C0	
0 0	Data In (DATI): a data word transferred to the master from the slave.
0 1	Data In Pause (DATIP): DATI, followed by a DATO or DATOB to the same location.
1 0	Data Out (DATO): a data word transferred from the master to the slave.
1 1	Data Out Byte (DATOB): a data byte transferred from the master to the slave. The lower or upper byte is specified by address bit ⟨A00⟩.
Parity ⟨PB:PA⟩	These signals transfer UNIBUS parity information. They are enabled by the slave during a transfer of data to the master (DATI). PA is not currently used and is not asserted.
PB PA	
0 0	No error
1 0	Parity error
0 1	Reserved
1 1	Reserved
Master Synchronization (MSYN)	MSYN is asserted by the master. It indicates to the slave that valid control and address information (with data on a DATO or DATOB) is present on the lines.
Slave Synchronization (SSYN)	SSYN is asserted by the slave. On a DATO(B) it indicates to the master that it has clocked the Write data. On a DATI(P) it indicates that it has asserted Read data on the UNIBUS for the master.
Priority Arbitration Group	
Nonprocessor Request (NPR)	An NPR request is from an I/O device for a DMA transfer that does not require processor intervention.
Nonprocessor Grant (NPG)	NPG is the processor response to the NPR and indicates to the I/O device that the request is being honored.
Bus Request ⟨BR7:BR4⟩	A bus request from an I/O device is for an interrupt operation.
Bus Grant ⟨BG7:BG4⟩	Bus grant signals are the processor response to a bus request. Only the highest request receives a bus grant signal at one time.

Table 1-2 UNIBUS Signal Description (Cont)

Signal Line	Description
Priority Arbitration Group (Cont)	
Select Acknowledge (SACK)	SACK is asserted by a bus-requesting device which has received a grant. Bus control passes to this device when the current bus master completes its operation.
Bus Busy (BBSY)	BBSY indicates that the address and data lines of the UNIBUS are in use. BBSY is asserted by the bus master until its operation is completed.
Interrupt (INTR)	INTR is asserted (instead of MSYN) by the interrupting device that has received a bus grant signal and has become bus master. This informs the UBI and CPU that an interrupt vector is present on the data lines. INTR is cleared upon receipt of SSYN from the UBI at the end of the transaction.

NOTE
All UNIBUS signals are asserted at ground level (low) except for the grant signals ⟨NPG,BG7:BG4⟩ which are asserted at +3 V (high).

Initialization Group	
Initialize (INIT)	INIT is asserted by the UBI module when DCLO is asserted on the UNIBUS. It remains asserted for approximately 70 ms following the negation of DCLO.
AC Line Low (ACLO)	ACLO warns of impending power failure. ACLO initiates the power-fail trap sequence and may be issued in peripheral devices to terminate operations and save data in preparation for power loss.
DC Line Low (DCLO)	DCLO is available from each system power supply and remains clear as long as all dc voltages are within specified limits. If an out-of-voltage condition occurs, DCLO is asserted.

1.4 THE CMI

The CPU/Memory Interconnect (CMI) consists of 45 bidirectional lines that carry address, data, and priority arbitration between all subsystems on the backplane. The signals of the CMI are divided into four groups: timing, data/address and control, priority arbitration, and status. Figure 1-4 and Table 1-3 provide descriptions of the CMI signals.

1.4.1 CMI Transfer Formats

Information is transferred between subsystems on the CMI by two operations. Each operation consists of transmitting a separate format on the CMI data lines. A master subsystem gains access to a slave by transmitting the physical longword address of the slave in the CMI address format (Figure 1-5) and asserting the DBBZ level for one B CLK cycle.

Bits ⟨01:00⟩ of the physical longword address field are not meaningful because data on the CMI is longword-aligned. The position of a byte in the CMI data longword is the effective address of the byte in relation to the physical longword address. A longword (four bytes of data) is then transferred to or from the slave in the CMI data format (Figure 1-6) while the slave asserts DBBZ.

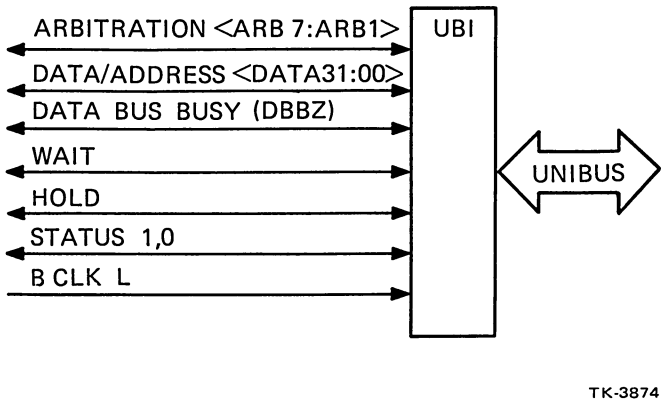


Figure 1-4 CMI Signals

Table 1-3 CMI Signal Description

Signal Line	Description
Timing	
B CLK L	B CLK L is generated by the CPU to synchronize system activity. A B CLK cycle is considered to be from one rising edge of B CLK L to the next. B CLK L is low for one-third of the cycle.
Data/Address and Control Group	
CMI Data ⟨31:00⟩	The CMI data lines are first asserted by a device that has assumed control as master. The master transmits control and address information to the slave (CMI address). The lines are then enabled for the transfer of data (CMI data). Bits ⟨01⟩ and ⟨00⟩ of the CMI address are ignored since four bytes (one longword) of data are represented on the lines. (See Section 1.4.1)
Data Bus Busy (DBBZ)	DBBZ is first asserted by the master for one CMI cycle while it places the CMI address on the CMI data lines. DBBZ is then asserted by the slave until data transfer is completed, except for a write operation where the slave is immediately ready to receive data.
HOLD	HOLD is used to temporarily suspend activity on the CMI by a device that requires more CMI cycles to complete a transaction.
WAIT	WAIT is asserted by a subsystem to initiate a processor interrupt. It is held until a Write Vector operation is performed.

NOTE
CMI data signals are asserted at +3 V (high); all other signals are asserted at ground (low).

Table 1-3 CMI Signal Description (Cont)

Signal Line	Description
Priority Arbitration Group	
⟨ARB7:ARB1⟩	An ARB level is assigned to each subsystem and is used to gain control of the CMI. If a higher priority bit is *not* set when a subsystem asserts its own priority bit, the subsystem assumes control of the CMI data lines. If a higher priority bit *is* set, the subsystem asserts its own priority bit to hold off lower priority subsystems until it gains control. Priority levels on the CMI are assigned as to the following devices:
ARB 7	RDM – highest priority
ARB 6	Reserved
ARB 5	Reserved
ARB 4	UBI (UBI 0)
ARB 3	MBA 0 (or optional UBI 1)
ARB 2	MBA 1
ARB 1	MBA 2
	CPU – lowest priority
Status Group	
STATUS 1,0	Status is transmitted by a slave to indicate the conditions under which data is returned to the master. Status bit combinations are defined as follows:
Status Bit **1 0**	
0 0	No response. Master attempted access to nonexistent memory (NXM) for read or write operation.
0 1	Data returned to master carried Uncorrectable Error (UCE).
1 0	Data was corrected.
1 1	Data had no errors.

The byte mask bits of the CMI address (Figure 1-5) designate which bytes are valid for transfer:

Byte Mask Bit	Byte(s) Valid for Transfer
Bit 31	Byte 3 valid
Bit 30	Byte 2 valid
Bit 29	Byte 1 valid
Bit 28	Byte 0 valid

The function code field (Figure 1-5) designates the operation that is being performed by the master:

Function bit 27	26	25	CMI Operation
0	0	0	Read
0	0	1	Read Lock
0	1	0	Read with Modify Intent
0	1	1	(undefined)
1	0	0	Write
1	0	1	Write Unlock
1	1	0	Write Vector
1	1	1	(undefined)

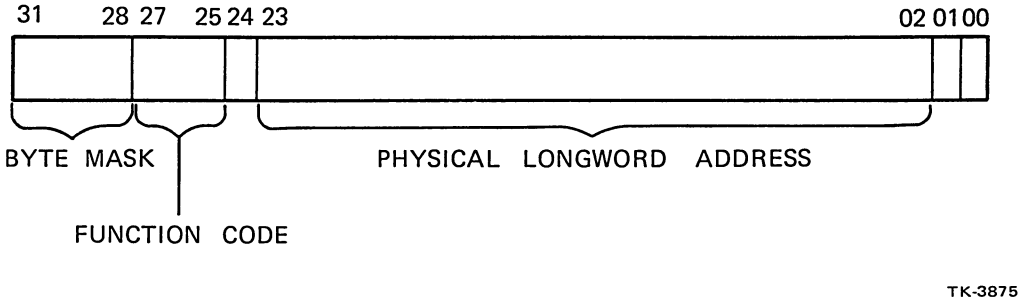


Figure 1-5 CMI Address Format

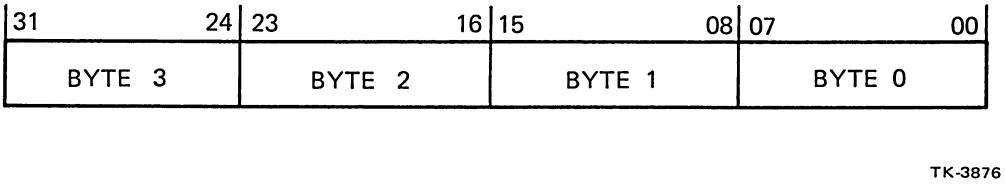


Figure 1-6 CMI Data Format

1.4.2 CMI/UNIBUS Data Transfers

Reference is frequently made to data transfers or to register contents as byte positions that correspond to specific data bits. Following is a list of those byte positions and the data bits to which they correspond:

Byte 0= Bits ⟨07:00⟩ of the UNIBUS data word or CMI data longword.

Byte 1= Bits ⟨15:08⟩ of the UNIBUS data word or CMI data longword.

Byte 2= Bits ⟨23:16⟩ of the CMI data longword.

Byte 3= Bits ⟨31:24⟩ of the CMI data longword.

When data transactions take place between the CMI and the UNIBUS, UNIBUS bytes ⟨1:0⟩ are read or written to CMI bytes ⟨1:0⟩, or to CMI bytes ⟨3:2⟩ as designated by UNIBUS (UB) address bit ⟨A1⟩. Transactions originated by the CPU always follow this convention and CMI data must be word-aligned to the UNIBUS.

For mapped transfers originated by the UNIBUS, the UBI byte offset bit, when enabled, causes data on even UNIBUS addresses to be transferred to or from odd CMI addresses. The UNIBUS data word (bytes ⟨1:0⟩) is then read/written to CMI bytes ⟨2:1⟩ instead of ⟨1:0⟩.

When the UNIBUS (UB) data word is designated for transfer with CMI bytes ⟨3:2⟩ and the offset is enabled, UNIBUS (UB) data byte ⟨0⟩ alone is transferred with byte ⟨3⟩ of the CMI data longword. UB data byte ⟨1⟩ extends beyond the boundary of the longword and is not transferred at the current CMI longword address. This is called the wrap condition. Reference is then made to the next longword address and UB data byte ⟨1⟩ is transferred with CMI byte ⟨0⟩.

1.5 FUNCTIONAL SECTIONS OF THE UBI

The CMI-to-UNIBUS Interface (UBI) is more than a simple bus converter. It adheres to the protocol of the CMI and the UNIBUS while monitoring and coordinating data transactions between them. B CLK L supplied by the CPU is used for all timing functions and synchronization. Figure 1-7 is a basic diagram of the main functional blocks that make up the UBI: the UNIBUS data path (UDP), address map (MAP), UNIBUS control (UCN), UBI control store and UNIBUS arbitrator.

1.5.1 UNIBUS Data Paths (UDP)

The UNIBUS data path section consists of four gate-array UDP chips. Each chip processes two-bit slices of a byte. The UDP section provides the necessary registers, gating, and alignment for data transfers between the UNIBUS which has 16 bits (two bytes) and the CMI which has 32 bits (four bytes). The UDP contains direct data path (DDP) gating, the buffered data path (BDP) registers, and buffered address (BAR) registers. It also contains a skew register to temporarily address or latch data information received from the CMI (CMI latch). The Received CMI Address Register (RCAR) stores CMI specified addresses for transfer to the UNIBUS address lines or to lines within the UBI.

1.5.2 Address Map (MAP)

The address map (MAP) is the facility that allows UNIBUS devices, which make sequential DMA transfers, to access noncontiguous pages of main memory. The 512 × 19-bit RAM is loaded by the software with the page frame numbers of main memory locations to be accessed, as well as the validity, byte offset, and data path information. UNIBUS NPR transfers take place on the direct data path or one of the three buffered data paths as designated by the map entry.

1.5.3 UNIBUS Control (UCN)

Control signal interpretations for transactions between the CMI and the UNIBUS are accomplished by the UCN section which is contained on a single gate-array chip. The UCN contains error and byte flags for each of the three buffered data paths. The byte flags are enabled to determine which bytes are valid for transfer to main memory. The error flags store nonexistent-memory and uncorrectable-error status. The UCN generates the CMI byte mask and function codes for UNIBUS transactions to main memory. In addition, UCN contains the slave control (SC) logic which provides CMI access for software intervention (access to the MAP or the error flags, or to initiate purge operation).

1.5.4 UBI Control Store

The UBI control store consists of a 256 × 24-bit PROM array with outputs clocked to a buffer register. In conjunction with BUT field gating in the UCN, it performs microsequences that execute and

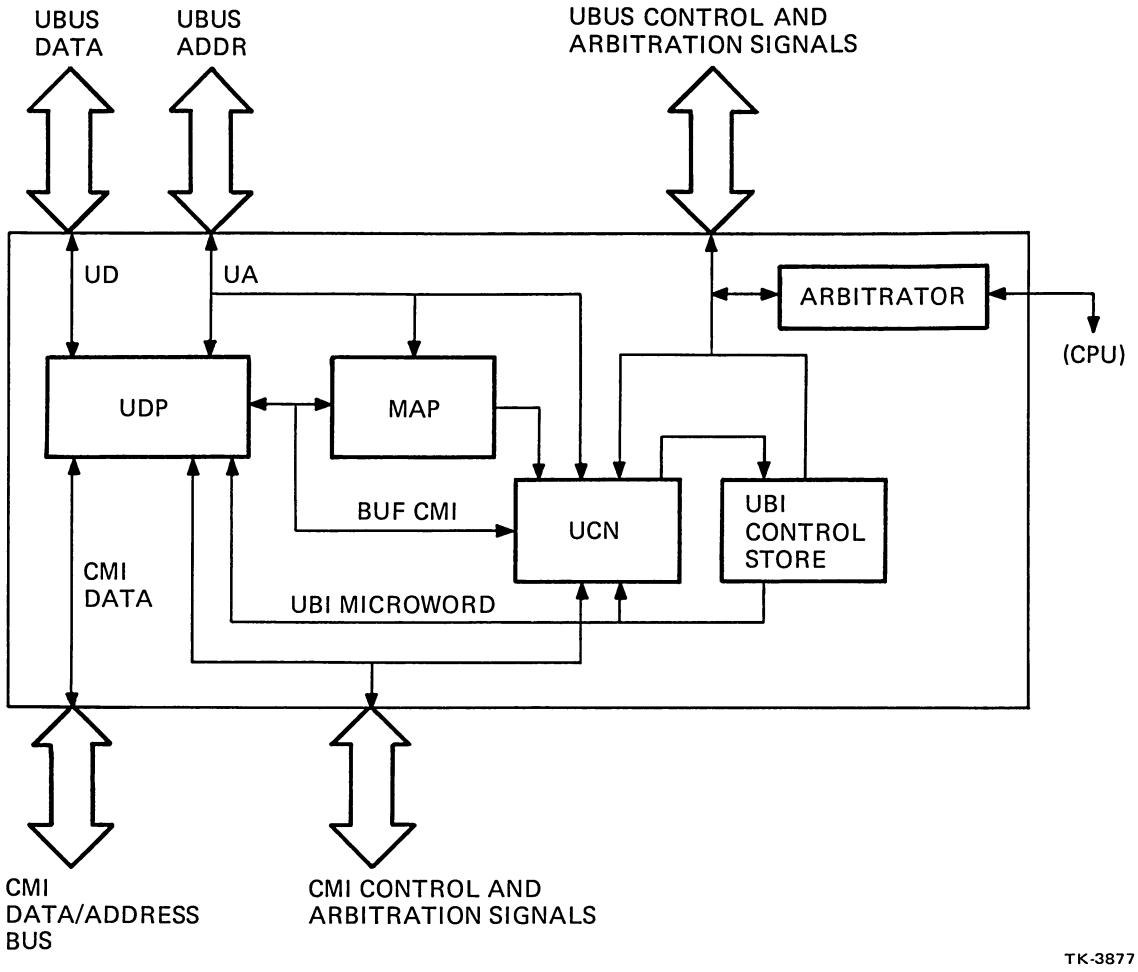


Figure 1-7 UBI Basic Diagram

direct UBI operations. Timing is provided by B CLK L supplied by the CPU. The UBI microword generates control signals for the UNIBUS, the MAP, and for priority arbitration on the CMI. It also generates fields that determine address and data gating through the UDP.

NOTE

The UBI control store is resident on the UBI module and should not be confused with the control stores of the CPU.

1.5.5 UNIBUS Arbitrator

The UNIBUS arbitrator selects the next UNIBUS master and generates a grant signal in response to an NPR or BR request. The CPU gains access to the UNIBUS through the arbitrator logic. BBSY is asserted when the CPU enables the CMI address longword for access to a UNIBUS device. Also, checks are made of processor status before a grant is issued to a BR interrupt requesting device.

1.5.6 UNIBUS Initialize

Initialization logic monitors the ACLO and DCLO signals on the UNIBUS. DCLO, driven by a UNIBUS power supply, initiates a processor microsequence to discontinue operations and assert the initialize level on the UNIBUS. This also clears logic and devices on the UNIBUS during a power-up

sequence. An ACLO condition asserts the SPFI signal (Sync Power-Fail Interrupt) to the INT chip. This generates a power-fail interrupt to prepare for loss of power.

1.5.7 UNIBUS Exerciser/Terminator (UET)

The M9313 UET module terminates the open-collector lines of the UNIBUS. It also contains registers and features that allow the diagnostic software to perform checks and exercise UNIBUS functions. A UNIBUS device need not be present to make use of these features. Refer to Appendix A for I/O and control information.

1.6 SYSTEM FUNCTIONS

Several CPU logic functions are also resident on the UBI module. CPU communication with major sections takes place on the W-Bus/W-CTRL field of the CPU microprogram. They are independent of the CMI, the UNIBUS, and the UBI control store. Functional descriptions on the following are provided in the VAX-11 KA750 Central Processor Technical Description.

- System Interrupts (INT)
- Console Interfaces (CON)
- Time of Year (TOY) Clock
- System Logic and Gating

1.6.1 System Interrupts (INT)

The single INT gate-array chip monitors microtrap and machine-check signals that generate processor interrupts. It also contains priority gating used by the UNIBUS arbitrator. Interrupts are generated for the following conditions:

- System Power Fail
- Write Bus (W-Bus) Error
- Corrected Memory Data
- Interval Timer
- Console Devices (Console Terminal, TU58)
- UNIBUS Interrupts

1.6.2 Console Interface (CON)

Two CON gate-array chips contain the serial/parallel, transmit and receive logic for the console terminal and cassette tape drive. A crystal oscillator drives the baud rate generator which is set by jumpers on the backplane.

1.6.3 Time of Year (TOY) Clock

The Time of Year clock consists of a 32-bit counter, incremented every 10 ms by a divide-by-ten network, driven by a 1-Khz oscillator. The offset memory is loaded with a preset value and the counter is cleared when the clock is first enabled. The offset value and counter contents are read by the CPU microcode and combined for a final value. Battery backup is provided to maintain clock operation for up to four days.

1.6.4 System Logic and Gating

CPU related logic on the UBI is covered in detail in the hardware handbook and in sections of the VAX-11 KA750 Central Processor Technical Description which apply to its complete functions:

- SID (System Identification) register – A set of eight, read-only gates are read by the CPU microprogram (CCS) on bits ⟨23:16⟩ of the W-Bus to construct the SID longword. The inputs are configured to a code that reflects the current system hardware revision level set by backplane jumpers.

  When completed, the hardware revision level occupies bits ⟨7:0⟩ of the SID longword; the firmware revision level occupies bits ⟨15:08⟩.

- FORCE TB PE and FORCE CACHE PE, driven from a decoder by the MISC field of the processor, are used by the diagnostic software to test parity-checking logic of the translation buffer and cache memory.

- Other logic is concerned with the return from microtrap (RTUT) function and with decoding for the distributed parity checking for the CCS, CS BUS ⟨4:0⟩ and FPA ⟨3:0⟩ field bits.

1.7 HARDWARE DESCRIPTION

The UBI is an extended length, hex-size module with a dedicated slot in the VAX-11/750 system backplane. It contains a total of eight gate-array chips:

- UDP (4)
- UCN (1)
- INT (1)
- CON (2)

UBI, as the UBI control store memory, also contains six 256 × 4-bit PROM chips blasted to standard matrices.

CHAPTER 2
FUNCTIONAL DESCRIPTION

2.1 GENERAL

Chapter 2 is a functional description of the UBI. The text provides descriptions and explanations of major circuit functions at the block diagram level.

2.2 UBI SUMMARY

The UBI provides for UNIBUS communication with the processor and main memory. It also provides the means for software intervention for monitor and control of UNIBUS activities.

2.2.1 Data Transactions

The resident UBI microprogram coordinates data transfers between the UNIBUS and the CMI.

CMI (CPU) initiated transactions:

- Read or write data to a UNIBUS I/O device register

UNIBUS initiated transactions:

- Read or write data to main memory
- Interrupt processor and write vector address

The CPU has direct access to the UBI for software intervention. These register transfers are independent of the microprogram:

- Read or write data to the MAP
- Read contents of a CSR (command/status register)
- Write a 1 to clear a flag in a CSR
- Write a 1 to initiate the purge operation for a buffered data path (BDP).

2.2.2 UBI Capabilities

The UBI has capabilities to support the following services for UNIBUS DMA (direct memory access) devices:

1. Transfer of data between the 16-bit UNIBUS and the 32-bit CMI using one direct and three buffered data paths
2. Address mapping between the 18-bit UNIBUS address and the 24-bit physical address of the CMI

3. Control/status monitor for all data paths

4. Data transfers to ascending or descending addresses in discontiguous pages of main memory

5. Odd-address (OFFSET) access to main memory space.

2.2.2.1 Buffered Data Paths (BDP) – One of three buffered data paths is selected on UNIBUS DMA transfers to accommodate:

1. High performance UNIBUS devices

2. Devices that generate sequential addresses

3. Two UNIBUS data word transfer operations to generate a single longword transfer on the CMI.

2.2.2.2 Direct Data Path (DDP) – The direct data path is selected on UNIBUS DMA transfers for:

1. Low performance devices
2. Devices that generate nonsequential addressing
3. Transfer operations by more than three devices
4. Devices that generate locked (DATIP) transfers.

2.3 UBI FUNCTIONAL BLOCKS

Figure 2-1 is a block diagram of the UBI that illustrates all functional blocks and major lines of communication:

1. UNIBUS Data Path (UDP)
2. Address Map (MAP)
3. UNIBUS Control (UCN)
4. UBI Control Store
5. UNIBUS Arbitrator

The UCN and UBI control store interact to provide the microsequencing for all UBI activities. Each block has its own unique set of functions and is described separately. However, descriptions of interaction between blocks is provided to clarify the contributions each makes to UBI functions. Access to the chips that make up the blocks is gained by bidirectional tristate ports. When neither the high nor low state is asserted by the drivers, the high-impedance (Hi-Z) off state exists.

2.3.1 UNIBUS Interface to the UBI

Figure 2-2 illustrates the interface between the UNIBUS open collector lines and the tristate lines of the UBI. Address bits ⟨17:02⟩ are transmitted to the UNIBUS from the BUF CMI port of the UDP. Received UNIBUS even addresses are gated directly to the UB address port of the UDP. An address is gated from the adder when the UNIBUS data word crosses the CMI data longword boundary described in Sections 1.4.1 and 1.4.2. The adder provides an increment of one at UB address bit ⟨02⟩. This is an effective increment of four to the physical longword address.

Two 2-bit fields from the UBI control store control gating of the UB address transceivers and mixer and the UB data transceivers. Tables 2-1 and 2-2 list these bits and their corresponding functions. The idle state value of both fields is 10 (receive UB address and data).

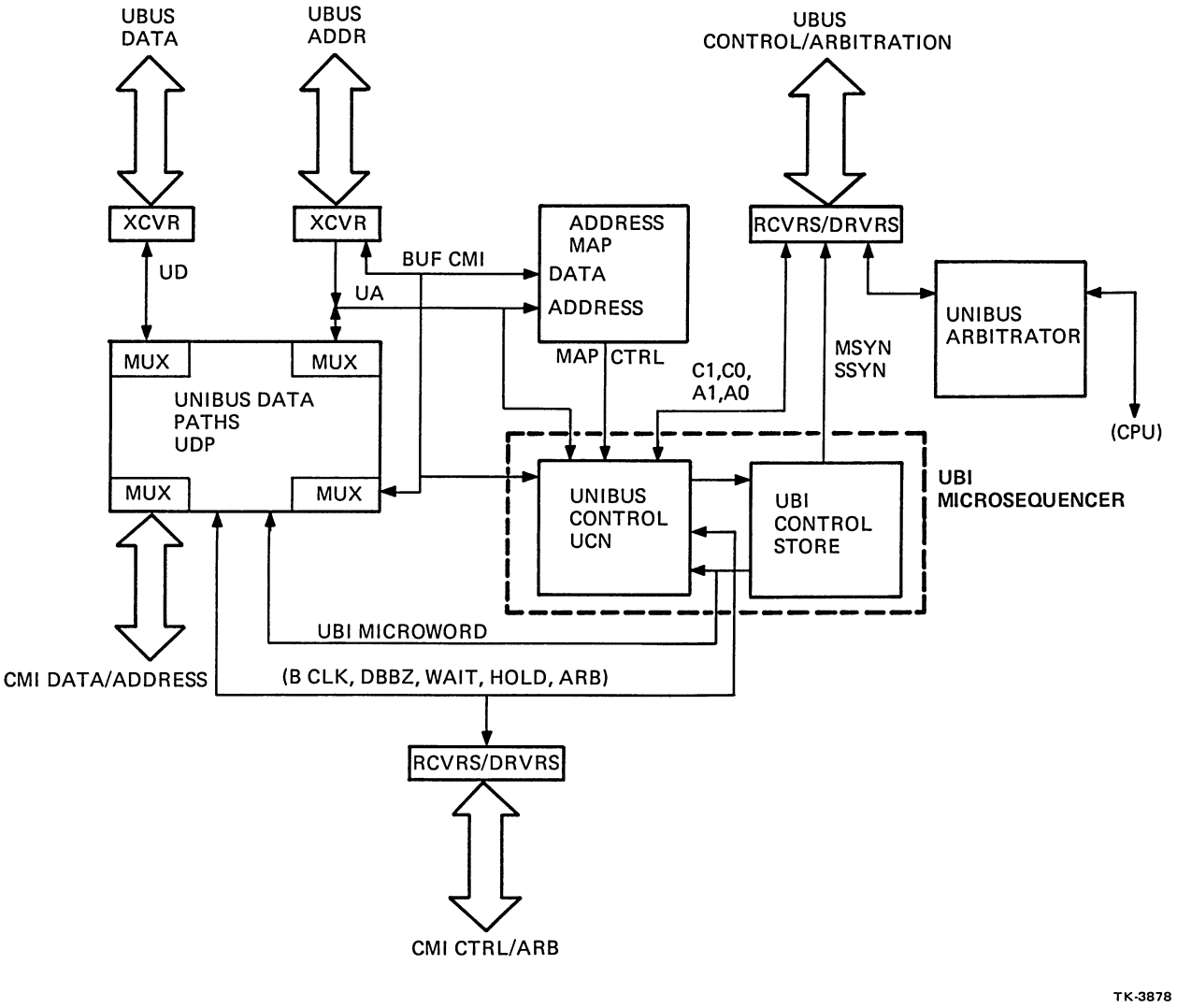


Figure 2-1 UBI Block Diagram

2.3.2 UNIBUS Exerciser/Terminator (UET)

The M9313 UET module is inserted in the UNIBUS OUT slot of the last device on the UNIBUS. Its features provide for:

1. Termination of the open-collector UNIBUS signals
2. Diagnostic exercise of:
   a. NPR data transfer capability
   b. BR interrupt capability
3. Optional 4K × 8-bit PROM for special applications.

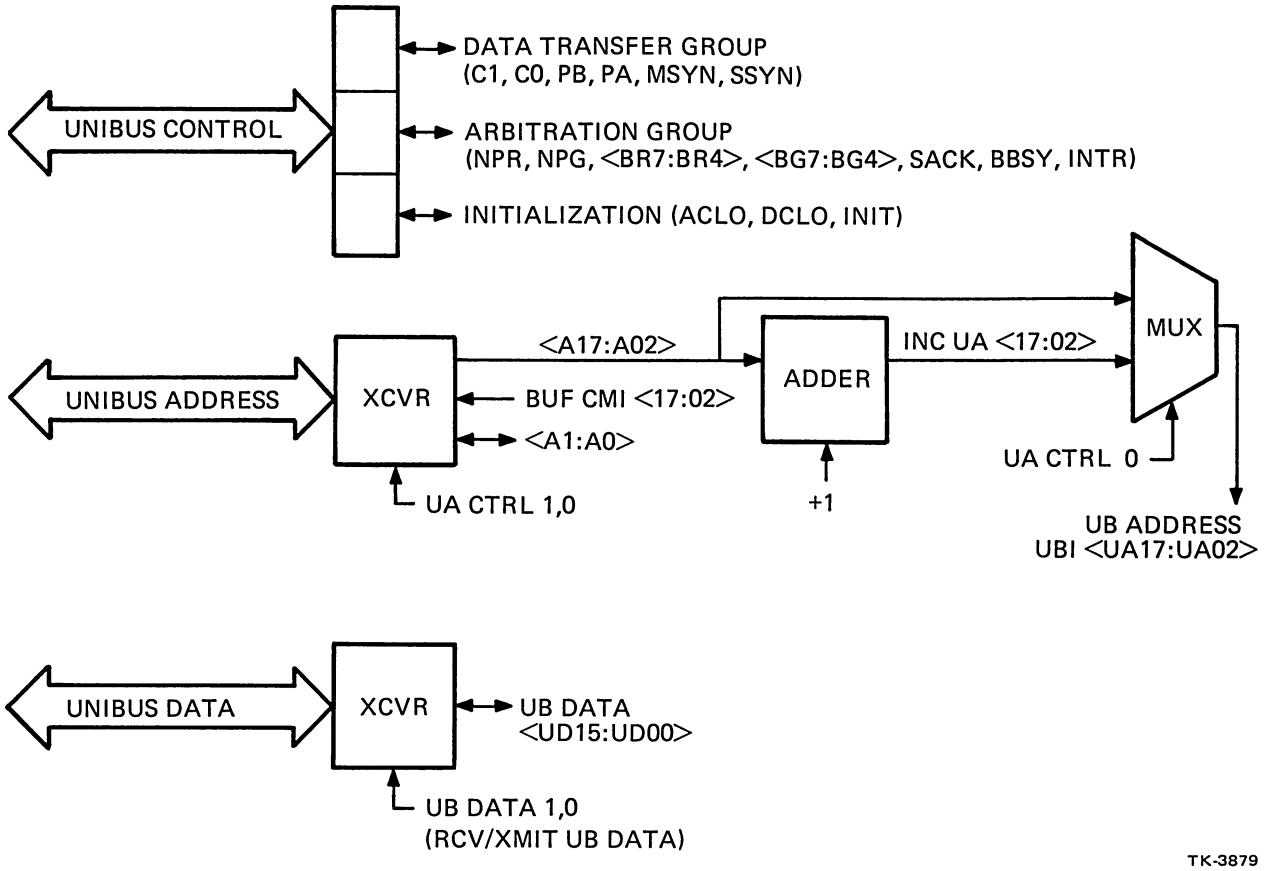


Figure 2-2 UNIBUS Interface to UBI

Table 2-1 UA CTRL Field Bits

UA CTRL Bit 1	UA CTRL Bit 0	Function
0	0	Enable BUF CMI to UB address lines
0	1	OFF (tristate port = Hi-Z)
1	0	Receive UB address
1	1	Receive and increment UB address

Table 2-2 UB DATA Field Bits

UB DATA Bit 1	UB DATA Bit 0	Function
1	0	Receive UB data
0	1	Transmit UB data
0	0	OFF (Hi-Z)
1	1	Transmit UB data; disable PB

2.4 UNIBUS DATA PATHS (UDP)

Figure 2-3 is a block diagram of the UDP section of the UBI. All address and data information is routed through the UDP which provides the necessary registers and gating for data alignment between the CMI and UNIBUS. Access to the UDP is made by four bidirectional tristate ports.

1. The UB data port for UNIBUS data
2. The UB address port for UNIBUS address
3. The CMI data port for CMI address and data
4. The BUF CMI port which makes CMI data information available to the UBI.

Four types of registers control address and data in the UDP.

1. Three buffered data path (BDP) registers each hold four bytes of data (CMI data bits ⟨31:00⟩).
2. Three buffered address registers (BARs) hold the addresses of data in the BDP registers (UNIBUS address bits ⟨17:02⟩).
3. Received CMI address register (RCAR) holds CMI address bits ⟨23:02⟩.
4. UNIBUS data latch (UB data latch) is part of the UB data port (UNIBUS data bits ⟨15:00⟩).

Two sets of multiplexer gating steer data to or from the UB data port:

1. Byte swap mux enables UB data word to the upper and lower word of the CMI data longword at the CMI data mux and BDP register inputs.
2. UB data byte select mux enables upper and lower words of the CMI data longword to inputs of the UB data latch.

For UNIBUS-initiated offset transfers, bytes of the UNIBUS data word are swapped as they are clocked to or from the UB data port.

Comparator gating is provided for CMI address longword (CPU) access to:

1. UNIBUS address space (ADDU)
2. CMI/UNIBUS interface address space (ADDC)
   a. Control/status registers (CSRs)
   b. Address map (MAP)

The match signal, when driven by the UDP, indicates to the UBI microprocessor whether UNIBUS data is in the same or in a different longword address as that stored in the BAR register for the selected BDP.

2.4.1 UDP Latch Registers

Data latches in the UDP are the feed-through type, consisting of NAND gates connected back to back. The latch outputs follow the input levels as long as the enabling inputs are open. Data on the input lines must first settle; the enabling signals then close to latch the data.

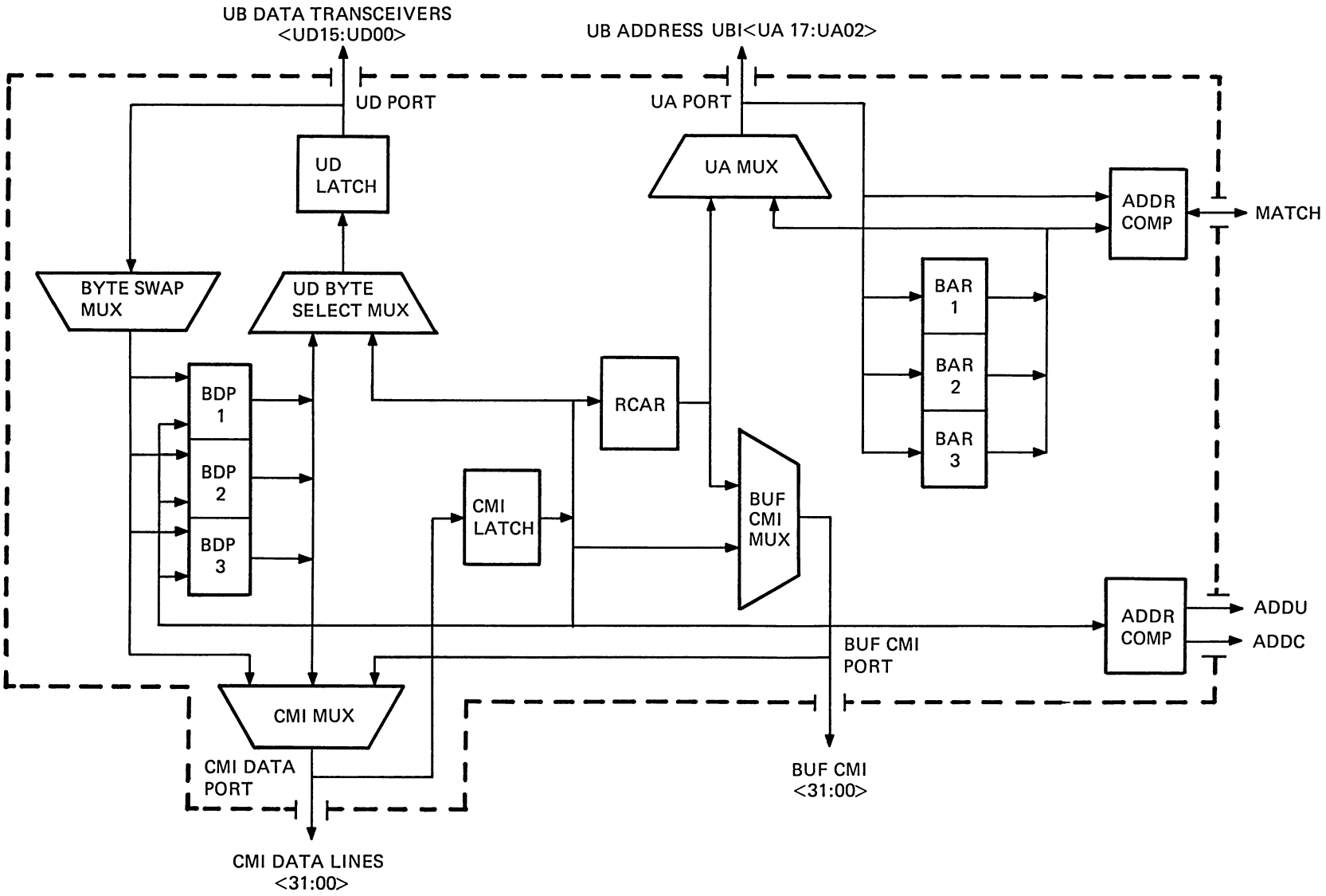


Figure 2-3 UDP Data Flow

2.4.1.1 UB Data Latch – UNIBUS data is gated through the UB data port which also serves as the UB data latch. This is possible because the receiver outputs are also connected to one of the input legs of the UB data drivers. The receivers are always enabled for received data; transmit data is clocked while the drivers are enabled. Data remains latched as long as the enable inputs to the drivers are active.

2.4.1.2 CMI Latch – All UDP address and data information received from the CMI data lines is clocked from the CMI latch. Output levels of the CMI latch follow levels from the CMI data lines while the B CLK L signal is high. When B CLK L goes low, data is latched to ensure that received data remains stable on the inputs of receiving registers. When B CLK L goes high, data is clocked into the receiving register and the CMI latch is open to new data.

2.4.2 Data/Address and Control Flow

All data and address information generated during CMI/UNIBUS transactions is either stored in or is gated through the UDP. The following descriptions are an introduction to UBI operations illustrating UDP data flow (refer to Figure 2-3).

2.4.2.1 CMI Initiated Transactions – A master device (the CPU) initiates a transaction on the CMI by transmitting a CMI address as described in Section 1.4.1.

1. CMI address is received by the CMI latch of the UDP.

    a. ADDU or ADDC level from the UDP is true.
    b. Byte mask and function code (CMI address bits ⟨31:25⟩) are clocked from the BUF CMI port to the BUF CMI latch in the UCN for decoding to UNIBUS control signals.
    c. Physical longword address (CMI address bits ⟨23:02⟩) is clocked to the RCAR.

2. When transaction is made to UNIBUS address space (ADDU is true):

    a. RCAR contents (device code of UNIBUS register or slave device) are enabled to the UNIBUS address lines from the BUF CMI mux.
    b. CMI data port transfers data directly to or from the UNIBUS data lines through the UB data port. The UNIBUS data word is gated between the upper or lower word of the CMI data longword as specified by the byte mask decoding of the UCN. This decoding drives the byte swap and byte select multiplexer gating.

NOTE

Much of the data lines and gating used by the DDP are enabled for CMI transfers with the UNIBUS. However, MAP control bits (valid, offset, and data path select bits) are disabled during CMI- initiated transfers. The address MAP does not apply, nor do any rules for UNIBUS-initiated, mapped transfers to the data paths.

3. When transaction is made to UBI address space (ADDC is true):

    a. RCAR contents (access to a MAP address or a CSR) are enabled to the UBI from the UB address port.
    b. CMI data port, 32 bits, is enabled through the BUF CMI port for transfer of applicable MAP or CSR bits.
    c. UBI microsequencer is not initiated. The transaction is accomplished by stepping of the slave control (SC) logic of the UCN.

4. A purge operation is initiated when CMI data bit ⟨00⟩ is written as a 1 to a CSR.

    a. Contents of the BAR register for the selected buffered data path are gated through the UB address port for access to the MAP (for address translation) and generation of the CMI address.
    b. Data is transferred from the selected BDP register to the CMI data port.

CMI byte mask and function code are interpreted by the UCN to generate equivalent UNIBUS operations:

CMI Operation	UNIBUS Equivalent
Read	DATI
Read Lock	DATIP
Read with Modify Intent	DATIP
Write	DATO(B)
Write Unlock	DATO(B)

2.4.2.2 UNIBUS Initiated Transactions – Transactions initiated by a UNIBUS master device are synchronized by the UNIBUS arbitrator section illustrated in Figure 2-1. The UBI performs an initial set of responses when access is required to the CMI.

1. UNIBUS control bits are decoded by the UCN to generate byte mask and function code.

2. UNIBUS address bits access the MAP to generate physical longword address, select data path, and determine validity and offset.

3. The UBI arbitrates for the CMI data lines and uses the above information to transmit the CMI address and gain access to main memory. This is the initiation of a CMI read or CMI write operation to main memory.

The sequence of UNIBUS NPR operations is determined by the type of data transfer operation taking place. Those operations are listed below.

Write (DATO) operation on a BDP:

1. The first UNIBUS NPR data word is transferred to the BDP register from the UB data port and gated from the byte swap multiplexer.

2. The UNIBUS address of the first data word (bits ⟨17:02⟩) is clocked into the BAR register from the UB address port.

3. SSYN is returned to the device and the transaction is completed.

4. The second UNIBUS NPR transfers the next sequential data word to the BDP register.

5. The UNIBUS address of the second data word (bits ⟨17:02⟩) is compared with that contained in the BAR register for a match, and is translated by the MAP to physical CMI address space. (No match indicates that the second data word is not within the same longword address as the first. Separate write operations take place to their respective longword addresses.)

6. UBI transmits a CMI address to access main memory and initiates a write operation on the CMI.

7. UBI transmits a CMI data longword to main memory, gating BDP register contents from CMI mux.

8. SSYN is returned to the device.

Read (DATI) operation on a BDP:

1. The first UNIBUS NPR request generates a UBI read operation which transmits the CMI address to gain access to main memory.

2. The CMI data longword, four bytes of data, is returned from main memory and clocked to the BDP register from the CMI latch.

3. The first word of BDP register is gated from byte-select multiplexer to UNIBUS data lines. SSYN is returned completing first UNIBUS request.

4. The second sequential UNIBUS NPR transfers a second word from BDP register to the UNIBUS device.

Read (DATI) or Write (DATO) on the DDP:

A single UNIBUS NPR request transmits a CMI address for access to main memory.

1. For a write (DATO), a UNIBUS data word is enabled to both the upper and lower word of the CMI data longword. Transfer of correct bytes to main memory is decided by byte mask which is decoded from UNIBUS control bits and the offset bit from the MAP.

2. For a read (DATI), four bytes of CMI data (one longword) are returned from main memory. Gating of correct bytes through the UB data byte-select multiplexer is determined by the UNIBUS control bits and MAP offset bit.

UNIBUS control bits are interpreted by the UCN to generate equivalent UBI operations on the CMI:

UNIBUS Operation	**CMI Equivalent**
DATI	Read
DATIP (on DDP)	Read Lock
DATO(B)	Write (Write Unlock if following a DATIP)

2.4.2.3 Interrupts – A UNIBUS device capable of interrupting the processor is assigned one of the standard device vectors. Assigned trap and interrupt vectors occupy UNIBUS addresses 000 through 274 octal (000 – 0BC hexadecimal). Floating vectors are available from 300 through 774 octal (0C0 – 1FC hexadecimal). A UNIBUS interrupt takes place in the following sequence:

1. BR arbitration takes place.

2. The vector address is asserted on the UNIBUS data lines and INTR (instead of MSYN) is asserted.

3. The UBI inclusive-ORs an offset of 200 (hexadecimal) to the vector address and initiates the write vector operation on the CMI.

4. The UBI returns SSYN to the device.

5. Processor response consists of:

   a. Retrieving the base address specified for the system control block (SCB), from the SCB base register.
   b. Adding the vector address (with UBI offset) received from the CMI.

   The resulting physical address is a vector that contains the virtual starting address of the device service routine.

2.4.3 UDP Signals

The following table lists signals that control and direct the flow of address and data through the UDP.

Table 2-3 UDP Signal Description

Signal	Description
B CLK L	B CLK L clocks the inputs to the latches and ports as directed by the BDPC, PRTC, and SC inputs.
BDPC ⟨02:00⟩	BDPC (Buffered Data Path Control) from the UBI control store directs clocking of the UB data latch and the BDP and BAR registers.
PRTC ⟨02:00⟩	PRTC (Port Control) from the UBI control store directs input/output enable levels to the four tristate ports.
SC ⟨01:00⟩	SC (Slave Control) from the UCN selects data and address gating on CMI-initiated transactions to the UBI address space.
PREV DBBZ	PREV DBBZ indicates whether the previous CMI cycle did have DBBZ asserted or not.
DBBZ	DBBZ (Data Bus Busy), when asserted with PREV DBBZ not asserted (DBBZ not asserted on previous CMI cycle), indicates to the UDP that the present CMI cycle has CMI address asserted. The address information is clocked to the RCAR if the ADDC or ADDU level is true.
OFFSET	OFFSET, when enabled by a MAP address translation, causes UNIBUS data to be rotated one byte to the left when transferred from the UNIBUS. It is rotated one byte to the right when transferred to the UNIBUS from the UBI.
DP SEL ⟨01:00⟩	DP SEL (Data Path Select) enables desired data path registers. **DP SEL Bits** **1 0 Data Path** 0 0 Direct Data Path (DDP) 0 1 Buffered Data Path (BDP1) 1 0 Buffered Data Path (BDP2) 1 1 Buffered Data Path (BDP3)
⟨A1:A0⟩	⟨A1:A0⟩ from the UNIBUS address (UB address port) have the following functions:

Table 2-3 UDP Signal Description(Cont)

Signal Line	Description
	⟨A1⟩ specifies one UNIBUS word to be transferred to or from the upper or lower CMI longword. ⟨A0⟩ used on a DATOB transfer specifies transfer of the upper or lower byte (of the upper or lower word when used with ⟨A1⟩).
ID	ID (Identify) line on each of the four UDP chips is connected to decode CMI latch bytes 2 and 1 in the comparison logic for ADDU and ADDC outputs.
ADDC	ADDC (Address CMI/UB Interface) indicates that the CMI address specifies an address in the range of F30000 to F31FFF (hexadecimal).
ADDU	ADDU (Address UNIBUS) indicates that the CMI address specifies an address in the range of FC0000 to FFFFFF (hexadecimal).
Match	Match is wire-ORed open collector between the UDP chips and the UCN driven at different times. When driven by the UDP, match indicates to the UBI that the data address transmitted by the UNIBUS (bits ⟨17:02⟩) is the same as the address stored in the BAR. Match driven low by the UCN during arbitration indicates to the UDP chips when the UBI has gained access to the CMI. The CMI data port drivers are enabled with the MAP address translation from the BUF CMI port.

2.5 ADDRESS MAP (MAP)

Figure 2-4 is a block diagram of the address map (MAP) facility of the UBI. Bits ⟨17:09⟩ of the address specified by the device are used to reference a MAP location (RAM memory, 512 × 19-bits). The output of this location is used to specify the page frame number (PFN) of main memory to be accessed. This MAP output is concatenated with bits ⟨08:02⟩ of the UNIBUS address by gating to the BUF CMI lines of the UDP. (Also see Figure 2-6.) This becomes the physical longword address field transmitted with byte mask and function code from the UCN as the CMI address to access main memory.

2.5.1 MAP Access and Data

When a UNIBUS I/O device is initialized for NPR operations, it is loaded with word count (WC), bus address (BA), and command register information. Sequential MAP locations must also be loaded with the page frame numbers of physical main memory valid for access. MAP addressing bits ⟨17:09⟩ are driven directly from the UB address lines during DMA transfers. For CMI access to the MAP, the RCAR contents are driven to these lines from the UA port of the UDP. Information is transferred to or from the MAP on the BUF CMI lines. Figure 2-5 defines the bit field information (shown in Figure 2-4) loaded to the MAP by a longword transfer from the CMI.

The MAP control bits, valid, offset, and data path select, are normally enabled by the UCN. (MAP CTRL OUT EN level in Figure 2-4 is normally asserted.) They are disabled for interrupts and for a purge operation where the data path is selected by the UCN and address translation is provided by the MAP.

The MAP OUT EN level is driven by bit ⟨23⟩, the most significant bit of the UBI control store microword (BUF CMI bit, see Section 3.1.2). When enabled during a microroutine, the output of the MAP location selected by UB address bits ⟨17:09⟩ are asserted onto the BUF CMI lines to the UDP with UB address bits ⟨08:02⟩. This physical longword address is asserted onto the CMI data lines by the UDP to access main memory.

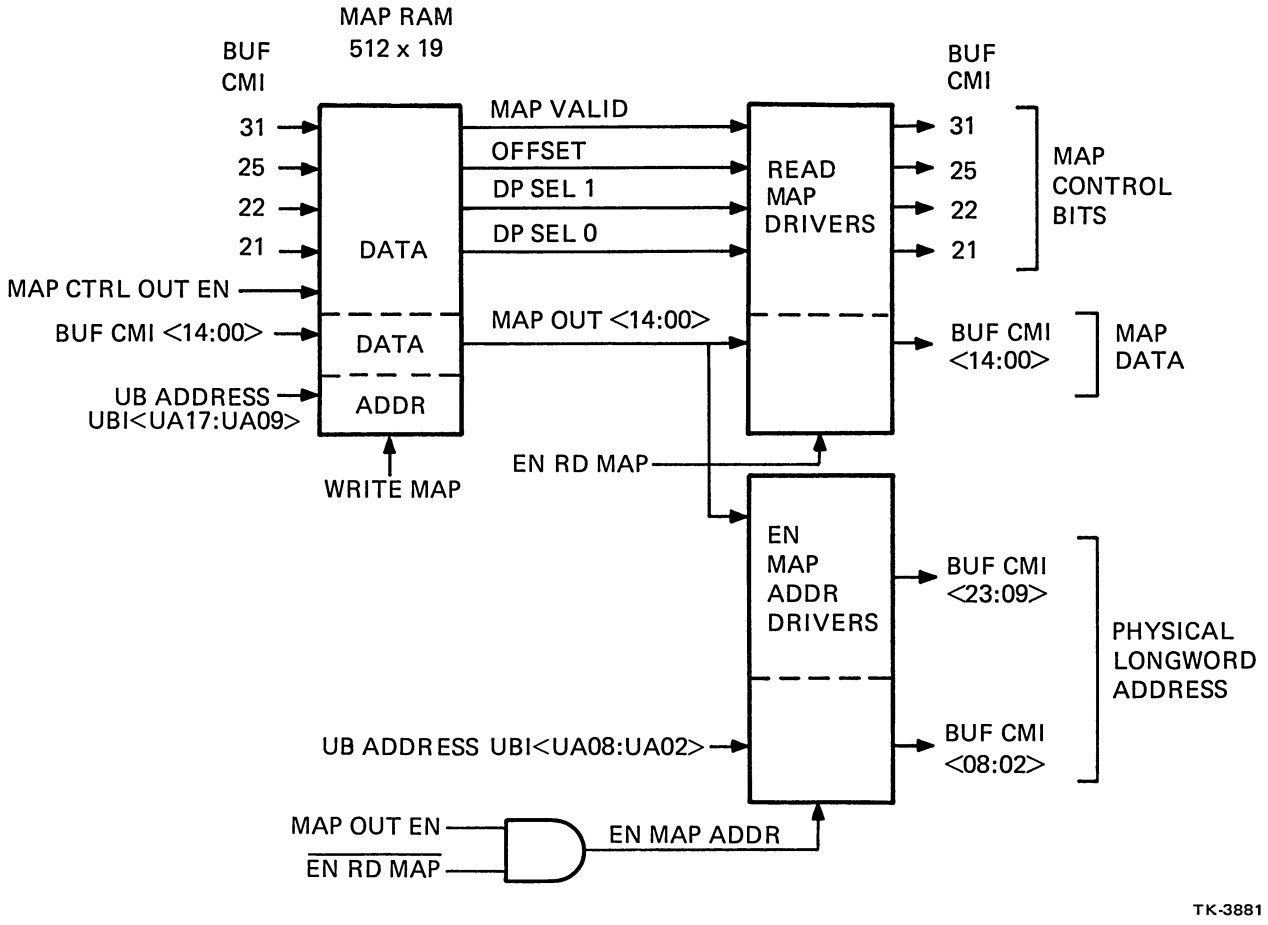


Figure 2-4 Address MAP

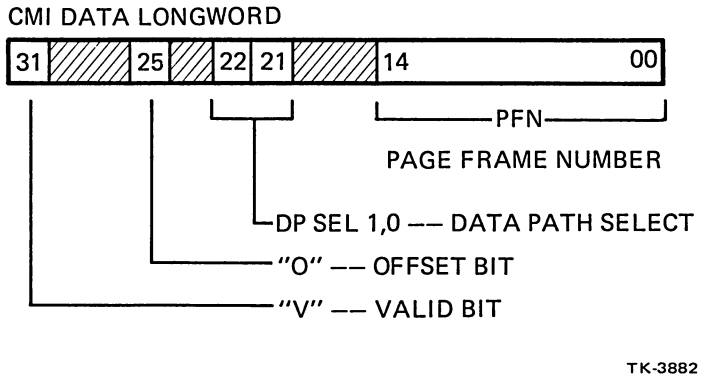


Figure 2-5 MAP Data

Page Frame Number – The page frame number specifies the physical address of 512 bytes of main memory. Enabled to the CMI as address bits ⟨23:09⟩ on a DMA transfer, the PFN is concatenated with UNIBUS address bits ⟨08:02⟩ to form the physical longword address field of the CMI address transmitted by the UBI.

Data Path Select Bits – The data path select bits select one of the four data paths:

0 0 – Direct Data Path (DDP)
0 1 – Buffered Data Path #1 (BDP1)
1 0 – Buffered Data Path #2 (BDP2)
1 1 – Buffered Data Path #3 (BDP3)

Offset – This bit allows UNIBUS devices to access main memory on odd-byte addresses. UNIBUS data is transferred to or from main memory locations one byte-address higher than specified by the UNIBUS device.

If a transaction that crosses page boundaries occurs with the OFFSET bit set, the data path number, offset, and valid bits must be identical in both MAP entries.

Valid – This bit, when set, processes the transaction to the CMI. When clear, the UBI ignores the receipt of MSYN, treating it as a NOP (no-operation).

NOTE

A UNIBUS device may perform data transfers between itself and another device on the UNIBUS. MAP addresses corresponding to those generated by the master device must have the valid bit clear to prevent UBI response.

2.5.2 UNIBUS MAP Address Translation

Figure 2-6 illustrates translation from the address specified by the UNIBUS device to the 32-bit CMI address transmitted on the CMI data lines.

Of address bits ⟨17:00⟩ driven by the UNIBUS device, bits ⟨A1:A0⟩ are routed to the UCN with control bits ⟨C1:C0⟩ to develop the byte mask and function codes. This activity is described in UCN Section 2.6.

Address bits ⟨17:09⟩ directly access a MAP location. The 15-bit output of the MAP location and bits ⟨08:02⟩ of the UNIBUS address are gated to the BUF CMI lines by the drivers shown in Figure 2-4. The UDP receives this output on the BUF CMI along with the byte mask and function code from the UCN. The UDP asserts all bits on the CMI data lines as the CMI address to gain access to main memory.

For each buffered data path, the UCN contains four byte flags which are set on UNIBUS DATO(B) transfers. They specify which bytes of the BDP register are valid for transfer to memory and are transmitted on the CMI as the byte mask. For UNIBUS DATI transfers, a byte mask of all 1s is transmitted. For a DATI to a BDP, its CD flag is set to indicate that received CMI data is available in the buffer (the byte flags remain clear).

UNIBUS functions are interpreted into CMI function code for equivalent operations:

UNIBUS Function	CMI Equivalent
DATI	Read
DATIP	Read Lock (DDP only)
DATO(B)	Write (Write Unlock if following a DATIP on the direct data path)

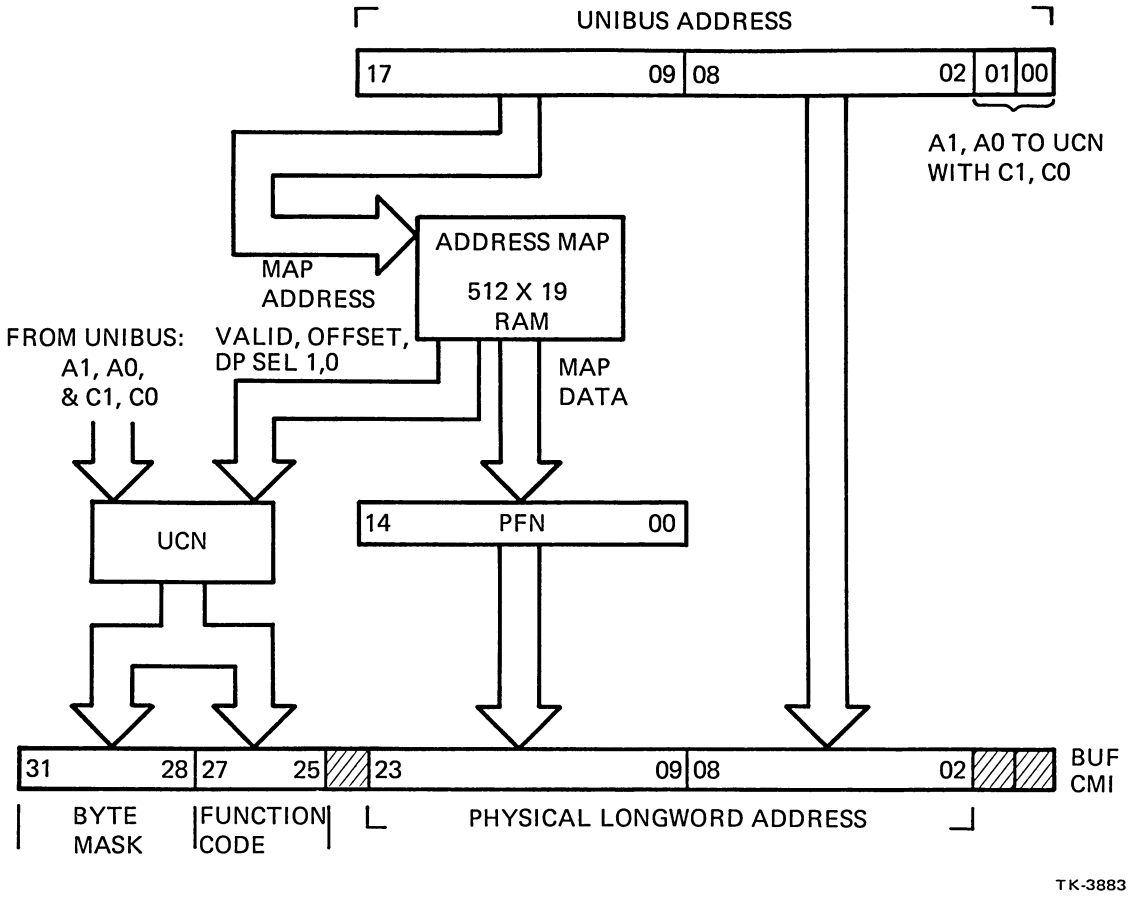


Figure 2-6 UNIBUS MAP Address Translation

2.5.3 CMI Physical Address Space

Figure 2-3 in Section 2.4, which illustrates data flow within the UDP, also shows the comparison logic that monitors bits ⟨23:16⟩ of the CMI latch. When an address transmitted by the CMI is in the range of FC0000 to FFFFFF (hexadecimal), the ADDU signal enables UBI logic which initiates a transaction on the UNIBUS. When the address is in the range of F30000 to F31FFF (hexadecimal), the ADDC signal enables operations within the UBI. Figure 2-7 represents blocks of physical address space allocated on the CMI. Over 15.7 megabytes of physical address space is reserved for MOS main memory. Addresses above main memory are reserved for subsystems and I/O device interfaces on the CMI. The highest addresses are reserved for UNIBUS devices.

2.5.3.1 UBI Address Space – The UBI is assigned 8 Kbytes of CMI address space for the MAP and control/status registers (CSRs). Figure 2-8 illustrates blocks of CMI addresses allocated for the standard and optional UBI. The base of the block is at address F30000 (hexadecimal).

2.5.3.2 CSR Data – A CSR exists for each of the buffered data paths. The flags are physically contained in the UCN chip. A CSR for a BDP consists of four bits of the CMI data longword as shown in Figure 2-9.

Bit ⟨00⟩ Purge Request (PUR) – Bit PUR normally reads as a 0. Writing a 0 produces no results. Writing a 1 produces results based on the contents of the BDP register:

UNIBUS data	Data is written to the CMI, byte flags are cleared to mark buffer empty.

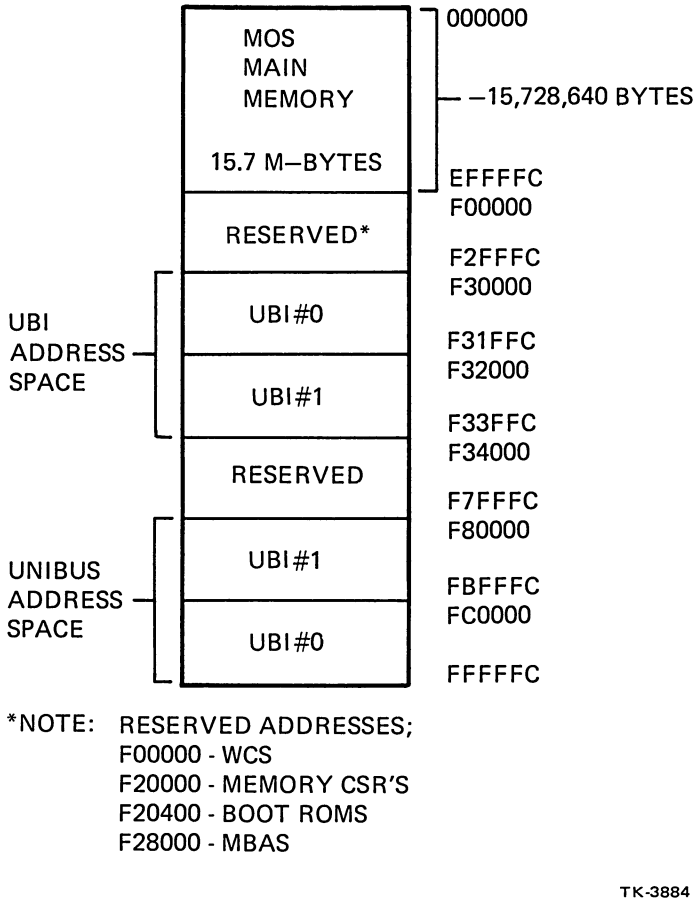


Figure 2-7 CMI Physical Address Space

CMI data	CD flag is cleared to mark buffer empty.
Buffer empty	No action occurs.

The purge request bit reads as a 1 until the purge operation is complete. This allows the software to determine when final DATO(B) data from the UNIBUS is in memory.

Bit ⟨29⟩ Uncorrectable Error (UCE) – Bit UCE indicates that UCE status was received from main memory. PB is asserted to the UNIBUS during the first read from that location. It is not asserted on subsequent reads from that location. A 1 is written to clear bit ⟨29⟩.

Bit ⟨30⟩ Nonexistent Memory (NXM) – Bit NXM indicates NXM status was received from main memory. SSYN is withheld from the UNIBUS device and further operations on this BDP are ignored until the bit is cleared by a write 1.

Bit ⟨31⟩ Error (ERR) – Bit ERR is the OR status of bits ⟨30⟩ and ⟨29⟩. This is a read-only bit.

2.5.3.3 CMI to UBI Address Space Mapping – When the CMI address accesses the UBI, ADDC is enabled by CMI bits ⟨23:16⟩ set to F3 (hexadecimal). CMI address bits ⟨11:02⟩ drive the UB address port lines of the UDP for access to a CSR or the MAP. The CMI to UBI address space mapping is illustrated in Figure 2-10.

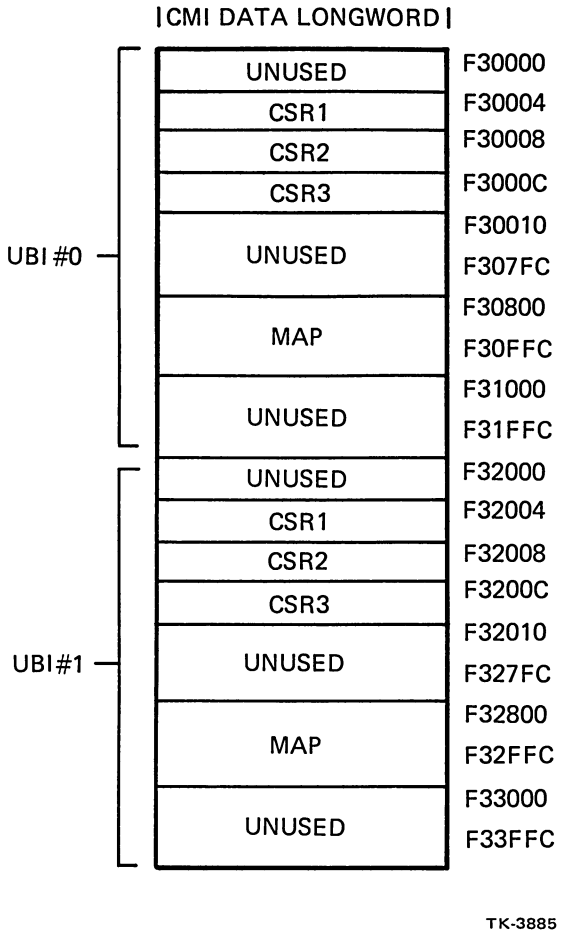


Figure 2-8 UBI Address Space

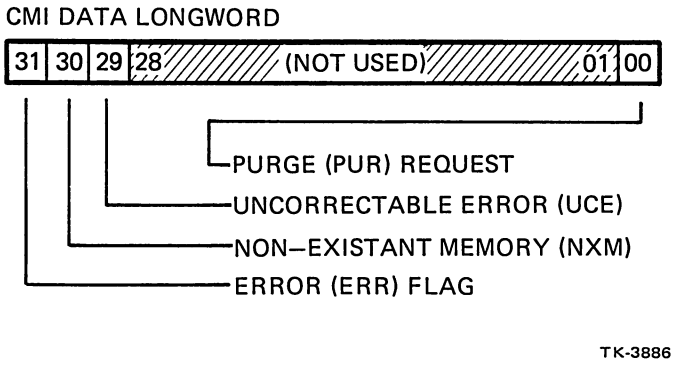


Figure 2-9 CSR Data

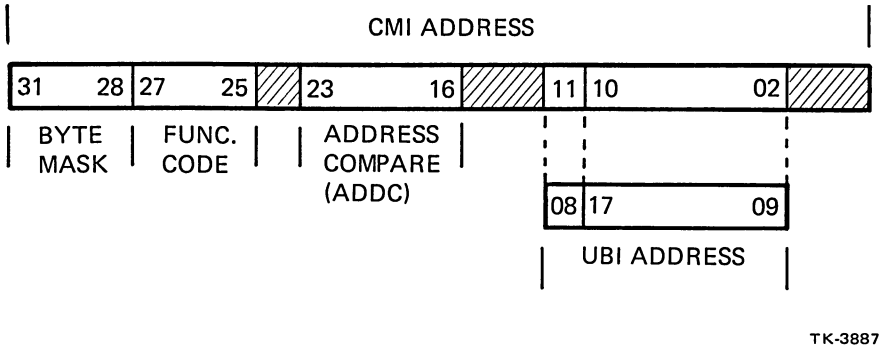


Figure 2-10 CMI to UBI Address Space Mapping

CMI address bits are decoded as follows:

- CMI bit ⟨11⟩ set to a 1 selects the MAP; CMI bits ⟨10:02⟩ select MAP locations.
- CMI bit ⟨11⟩ set to a 0 selects the CSRs; CMI bits ⟨04:02⟩ select specific CSR.

CMI Bits 04	03	02	Register
0	0	0	Unused
0	0	1	CSR #1
0	1	0	CSR #2
0	1	1	CSR #3
1	X	X	Unused

2.5.3.4 CMI to UNIBUS Address Space Mapping – Figure 2-11 illustrates mapping between the physical longword address of the CMI and the UNIBUS address lines which are driven from the BUF CMI lines of the UDP.

When the CMI address accesses an I/O device on the UNIBUS, CMI bits ⟨23:18⟩ are set from FC through FF (hexadecimal). ADDU is enabled and all rules of UNIBUS addressing apply.

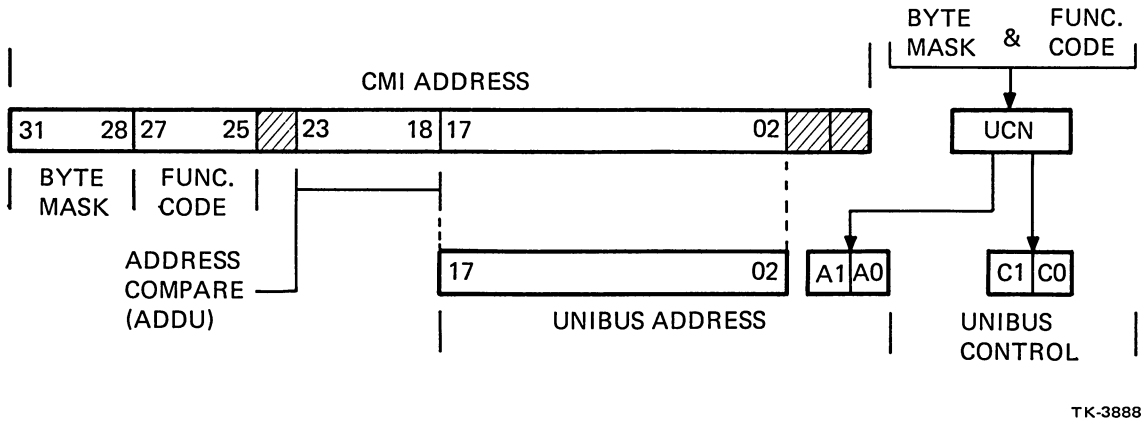


Figure 2-11 CMI to UNIBUS Address Space Mapping

2.6 UNIBUS CONTROL (UCN)

Figure 2-12 is a block diagram of the UCN section that consists of a single gate-array chip. The UCN defines operational conditions and directs UBI control store sequences to perform all data transactions. The UCN contains the following registers and gating:

1. Logic creates byte mask and function codes on the UNIBUS-initiated transactions to the CMI.

2. BUF CMI latch holds the byte mask and function code bits on CMI-initiated transactions to the UBI or UNIBUS.

3. Byte flags for each BDP, and Byte select gating for the DDP, indicate valid bytes on UNIBUS-initiated DATO(B) transactions. CD flag (CMI data flag) for each BDP indicates that DATI data is available in the buffer.

4. CMI-UNIBUS control signal interpretation and status response logic.

5. The CSR for each buffered data path consists of NXM and UCE error flags and the purge request bit.

6. Branch under test (BUT) field from UBI control store is decoded by the UCN to direct microsequencer operations by altering the NEXT address field of the UBI control store.

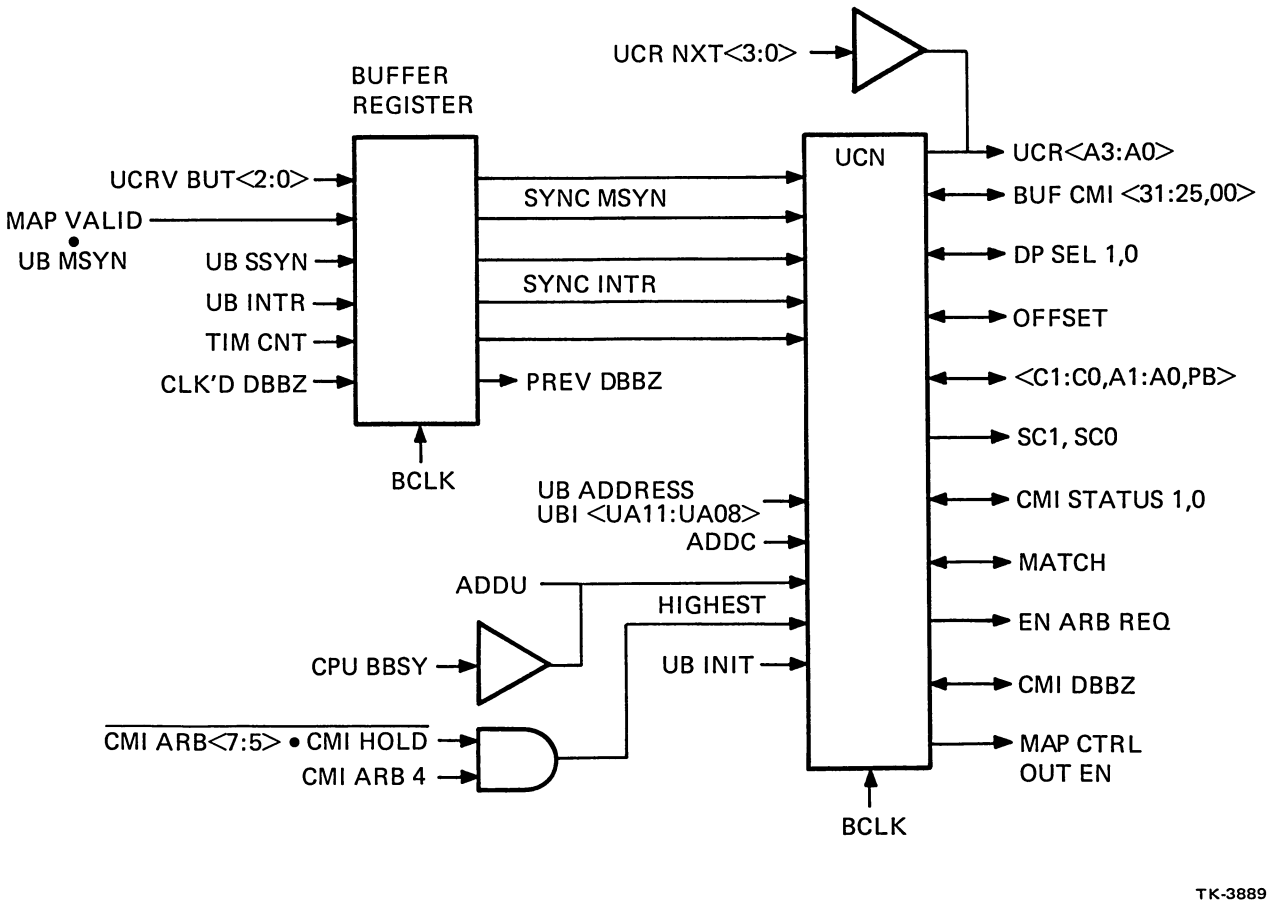


Figure 2-12 UNIBUS Control (UCN)

7. Slave control (SC) logic allows CMI access to MAP or CSRs.

8. Purge logic allows buffered data paths.

2.6.1 CMI Initiated Transactions

On CMI-initiated transactions to UBI address space, the CMI data longword transfers applicable bits with the MAP or with a CSR in the UCN. This activity, described in Section 2.5.3, is a register transfer accomplished by the slave control (SC) and no UBI microsequences are generated.

For CMI-initiated transactions to UNIBUS address space, the ADDU signal from the UDP is clocked to the latch ADDU flop in the UCN and BBSY is asserted on the UNIBUS by the arbitrator. The UBI microsequencer assumes control as master device on the UNIBUS.

2.6.1.1 CMI Address Cycle – Like other UBI logic and subsystems of the CMI, the UCN contains a PREV DBBZ flip-flop that retains the asserted or deasserted state of DBBZ from the previous B CLK cycle. Arbitration takes place during a cycle with DBBZ not asserted. The highest priority subsystem with an arbitration level asserted wins access to the CMI. On the following cycle, the subsystem asserts a CMI address with DBBZ. The combination of the PREV DBBZ flop cleared with DBBZ asserted indicates to all other subsystems that an address is present on the CMI. Only the CPU has access to UBI or UNIBUS address space. An access attempt by any other subsystem is ignored by the UBI.

2.6.1.2 CMI to UNIBUS Control Decode – Byte mask and function code bits of the CMI-initiated address cycle are received by the UCN on the BUF CMI lines from the UDP and clocked to the BUF CMI latch. As shown in Table 2-4, byte mask and function code bits are decoded to UNIBUS bits ⟨C1:C0⟩ and ⟨A1:A0⟩ to direct word/byte transfers between the CMI and UNIBUS.

Table 2-4 CMI to UNIBUS Control Decode

Function Code Bits			CMI	UNIBUS	Control Bits	
27	26	25	Operation	Operation	C1	C0
0	0	0	Read	DATI	0	0
0	0	1	Read Lock	DATIP	0	1
0	1	0	Read with Modify Intent	DATIP	0	1
0	1	1	(undefined)	–	–	–
1	0	0	Write	DATO(B)	1	(1)
1	0	1	Write Unlock	DATO(B)	1	(1)
1	1	0	Write Vector	(no response)	–	–
1	1	1	(undefined)	–	–	–

NOTES

The operation is always defined with respect to the master, in this case the CMI.

When a DATIP is generated, BBSY is asserted by the UBI until the end of the DATO(B) of the DATIP/DATO(B) sequence.

The choice of DATO or DATOB is made on the byte mask.

Table 2-5 illustrates the decode gating that translates CMI byte mask codes to UNIBUS control signals for a CPU write to the UNIBUS.

Table 2-5 CPU Byte Mask Write Codes

Byte Mask	A1	A0	C0	Data Size
0011	0	0	0	Word
1100	1	0	0	Word
0001	0	0	1	Byte
0010	0	1	1	Byte
0100	1	0	1	Byte
1000	1	1	1	Byte

For CMI-initiated transfers to the UNIBUS, the processor only produces a given set of byte mask combinations. For a read operation, a longword is always returned to the processor which selects the data it requires. Table 2-6 illustrates byte mask translation to UNIBUS control signals for a CPU read on the UNIBUS.

Table 2-6 CPU Byte Mask Read Codes

Byte Mask	A1	A0	C0	Data Size
1111	0	0	X	Word/Byte
1110	0	0	X	Byte
1100	1	0	X	Word/Byte
1000	1	0	X	Byte

X = 0 for normal read
X = 1 for read-modify or read-lock

2.6.1.3 UBI Response to UNIBUS Status – Depending on UNIBUS response to CMI-initiated transfers, the UBI returns the status to the CMI as shown in Table 2-7.

Table 2-7 UBI Response to UNIBUS Status

UNIBUS Response	UBI Status to CMI
No SSYN for 16 μS	NXM status (nonexistent memory)
DATO(B), SSYN returned*	No Error status
DATI(P), SSYN returned*	No Error status
DATI(P), SSYN returned†	UCE status (uncorrectable error)

*PB, PA = 00
†PB, PA = 10

2.6.1.4 Slave Control – The slave control consists of a 3-bit stepping register ⟨SST2:SST0⟩ clocked by B CLK L on a CPU reference to the MAP or a CSR (ADDC enabled). Data, address, and control gating within the UCN and UDP are redirected by the SST register and by the SC1 and SC0 signals driven from bits SST1 and SST0. UB address port drivers are enabled with RCAR contents to access MAP or CSR. Gating is also enabled to transfer MAP or CSR data between the BUF CMI and CMI data ports. For the idle state, SC1 H is high and SC0 H is low and SST ⟨2:0⟩ = 0's for normal CMI/UNIBUS microsequencer operations.

Slave control sequencing is discussed in more detail in Chapter 3.

2.6.2 UNIBUS Initiated Transactions

The UCN response to UNIBUS requests is to initiate microsequences that transfer data between the UBI and UNIBUS. A transfer of data between the UBI and the CMI is a separate operation, coordinated by the microprogram.

2.6.2.1 UNIBUS to CMI Control Decode – UNIBUS control bits are translated by the UCN to generate equivalent CMI functions. Table 2-8 shows the translation from UNIBUS control bits to CMI functions.

Table 2-8 UNIBUS to CMI Control Decode

UNIBUS Operation	CMI Operation	Function Code Bits 27	26	25
DATI	Read	0	0	0
DATIP	Read Lock (on the DDP)	0	0	1
DATO(B)	Write	1	0	0
DATO(B)	Write Unlock (when following a DATIP)	1	0	1
INTR	Write Vector	1	1	0

DATO(B) transfers on a BDP set byte flags in the UCN that generate the byte mask transmitted on the CMI. The same gating is true for DATO(B) transfers on the DDP. However, the results are transmitted as the byte mask during the CMI address cycle and no byte flags are set.

A byte flag is set as a byte of UNIBUS data is clocked to its corresponding byte position of the CMI data longword in the BDP register. Table 2-9 illustrates flag and byte clocking as selected by the UNIBUS control bits and the offset bit from the MAP. The decoded byte flags and function code are enabled on the BUF CMI lines to the UDP for transmission as part of the CMI address to main memory.

Table 2-9 UNIBUS Byte Mask Select

UNIBUS Operations	UNIBUS Control Bits C0	A1	A0	Offset	Set Byte Flags BF3	BF2	BF1	BF0
DATO	0	0	–	0	–	–	1	1
	0	1	–	0	1	1	–	–
	0	0	–	1	–	1	1	–
	0	1	–	1	1	–	–	*
DATOB	1	0	0	0	–	–	–	1
	1	0	1	0	–	–	1	–
	1	1	0	0	–	1	–	–
	1	1	1	0	1	–	–	–

Table 2-9 UNIBUS Byte Mask Select (Cont)

UNIBUS Operations	UNIBUS Control Bits				Set Byte Flags			
	C0	A1	A0	Offset	BF3	BF2	BF1	BF0
	1	0	0	1	–	–	1	–
	1	0	1	1	–	1	–	–
	1	1	0	1	1	–	–	–
	1	1	1	1	–	–	–	*
DATI	0	–	–	–	0	0	0	0 † (CD = 1)

*Wrap – The DATO(B) wrap microsequences first write existing buffer data to memory. The upper byte of a UNIBUS data word that crosses the CMI data longword boundary is then clocked to byte 0 of the BDP register and byte flag 0 is set. The UNIBUS address is incremented and clocked to the BAR register to specify the next physical longword address. Data may also wrap to the following page frame, crossing the page boundary. When this occurs, the MAP control bits (valid, offset, and data path select) of both page frames must have the same values or the results are unpredictable.

Byte flags are cleared during the UBI write operation when status is clocked from main memory and DBBZ deasserted.

†Although no byte flags are set, a byte mask of 1111_2 is transmitted for all DATIs. The CD flag (CMI data flag) is set when BDP contents are received from the CMI during a UBI read operation. The CD flag is used within the UCN, as are the byte flags, to define CMI/UNIBUS and purge operations. It is transparent to the software. The use of the CD flag is as follows:

CD = 1 – BDP contains data longword from the CMI as the result of a DATI from the UNIBUS. Byte flags ⟨BF3:BF0⟩ are cleared. A purge operation clears the CD Flag.

CD = 0 – BDP contains DATO(B) data from the UNIBUS if any byte flags are set. A purge operation initiates a transfer to the CMI and the byte flags are cleared.

All Flags = 0 – BDP buffer register is considered empty of data. A purge request initiates no operation.

2.6.2.2 UBI Response to CMI Status—CSR – The UBI receives status with the read data returned from main memory or when write access is attempted. The UBI responds by altering its activity to the UNIBUS. For a BDP, CMI status is clocked to the CSR. Table 2-10 shows the CMI status and correspondent UBI response.

Table 2-10 UBI Response to CMI Status

CMI Status	UBI Activity to UNIBUS
No Error or Corrected Data	SSYN is issued
NXM (Nonexistent Memory)	SSYN is withheld
UCE (Uncorrectable Error)	PB is asserted with SSYN

2.6.2.3 CMI Arbitration – A device initiates operations by raising its NPR or BR level and executing an arbitration cycle on the UNIBUS. If the UNIBUS transaction to the UBI requires activity on the CMI, the CMI ARB 4 level is generated by the microprogram and asserted on the CMI to hold off lower priority subsystems. When higher priority levels, CMI ARB ⟨7:5⟩, and HOLD are deasserted as

Figure 2-12 illustrates, the UBI is the highest requesting device. When the UCN detects the absence of DBBZ for one B CLK L cycle, the UBI has control of the CMI and executes the CMI address cycle described in Section 2.6.1. The address translation is asserted by the UDP, with DBBZ from the UCN, to select main memory.

2.6.3 Purge Response

The UCN monitors conditions that require responses from the purge logic. UCN also directs the microsequencer from the BUT gating.

2.6.3.1 Purge Request – A purge request is issued by the software service routine upon completion of data transfers by a UNIBUS device. The action taken by the purge logic depends upon the state of the byte flags for the final data remaining in the buffer. The following lists those states and the consequent action taken by the logic.

1. For DATI data remaining in the BDP register from the CMI, the CD flag is set to a 1. The byte flags are not set. The purge action clears the CD flag.

2. If the final DATO(B) transfer from the UNIBUS did not leave all byte flags set, a buffer "Not Full" condition did not cause the final CMI transfer to take place. The purge action initiates a UBI write operation to transfer the final data to memory and clear the byte flags.

3. If all flags are 0 at the end of UNIBUS DATI or DATO(B) operations, the BDP register is considered empty of valid data. No action occurs if a purge request is issued to the direct data path.

2.6.3.2 Auto Purge – Auto purge is the means by which the UBI may free a BDP register of DATO(B) data to accept further data without software intervention.

1. If a UNIBUS device clocks DATO(B) data to the BDP buffer and all byte flags are not set, the UBI write is not generated to transfer the data to memory.

2. If the device performs another operation to the BDP (DATI or DATO(B)) at a different longword address, the UBI write is generated to transfer the existing data to memory. This clears the byte flags leaving the register free for the second transaction.

2.7 UBI CONTROL STORE

Figure 2-13 is a block diagram of the UBI control store. Outputs of the 256 × 24-bit PROM array are clocked to the buffer register for the UBI microword by B CLK L.

Bits ⟨6:4⟩ of the NEXT (UCR NXT) field drive PROM address lines directly. The BUT field drives decode gating in the UCN which conducts microsequencer operations by altering UCR address bits ⟨3:0⟩. These bits are wire-OR'd with NEXT field bits ⟨3:0⟩ to provide steering of microsequencer addresses.

The power-up microword resides throughout the highest-order range of addresses. An address within that range is selected by initialize from the UNIBUS (UB INIT) when the machine is turned ON. Executing the power-up code directs the microsequencer to the idle address where it remains until microsequencer activity is produced.

Details of microsequencer operations are discussed in Chapter 3, Detailed Logic Description.

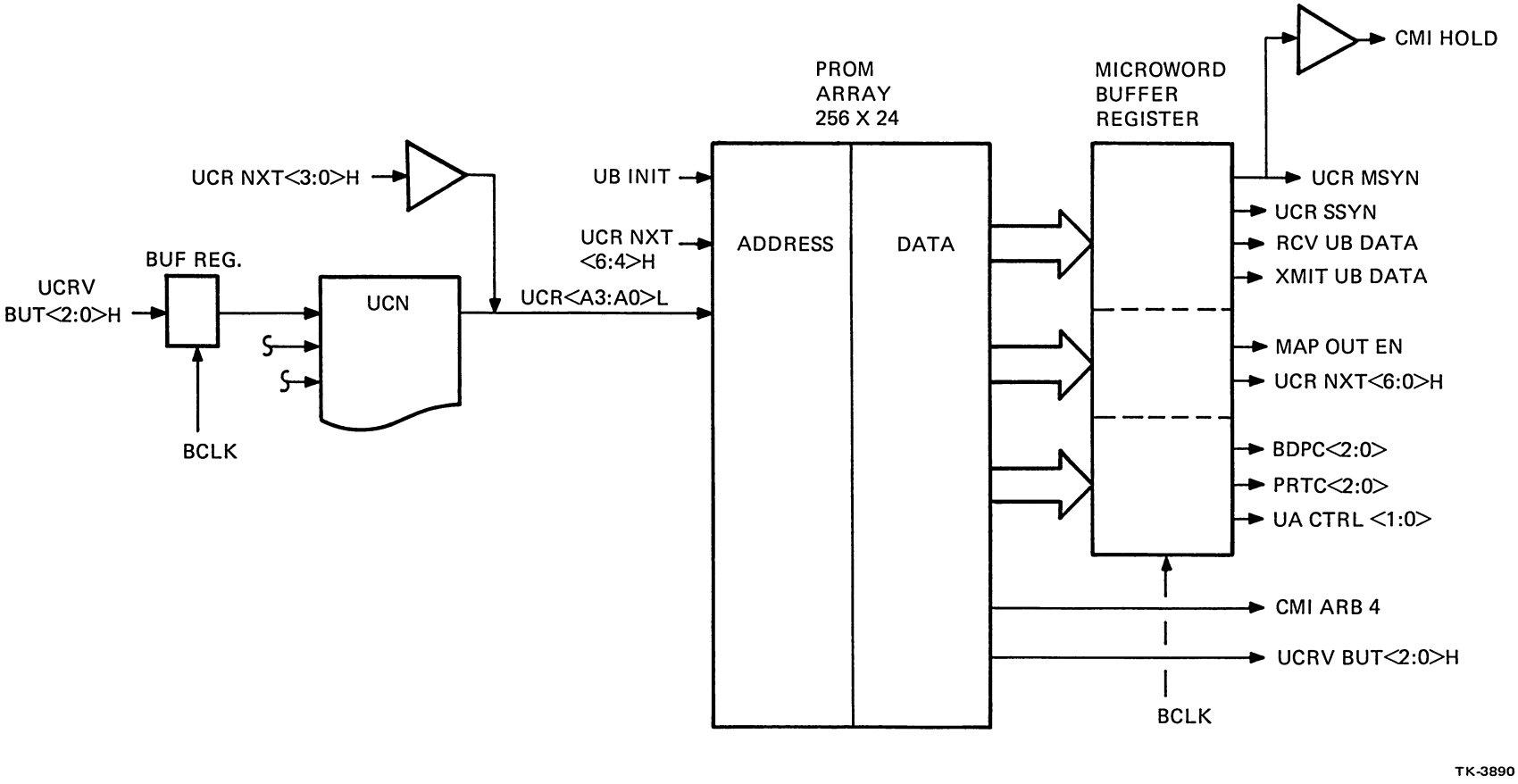


Figure 2-13 UBI Control Store

CHAPTER 3
DETAILED LOGIC DESCRIPTION

3.1 UBI MICROPROCESSOR

Chapter 3 provides a detailed description of UBI data transfer operations, the UBI microsequencer, and associated logic. Examples of data transfers are coordinated with flowcharts to illustrate clocking and gating of data within the UDP.

The UDP section of the UBI is described in Section 2.4. Figure 2-3, UDP Data Flow, is a recommended reference for Sections 3.2 and 3.3 which are concerned with data transactions. These sections and accompanying figures acquaint the reader with registers, ports, and multiplexer gating of the UDP. The reader should also be acquainted with the CMI transfer formats described in Sections 1.4.1 and 1.4.2.

3.1.1 Power Up and Initialize

Figure 3-1 illustrates the UBI control store power-up sequence generated by turning the system ON. The power-up code resides in high-order microaddresses ⟨80⟩ through ⟨FF⟩ (hexadecimal). An address within that range is selected by the UNIBUS-initialize level and clocked by B CLK L issued by the CPU. Once dc voltages stabilize, address ⟨8F⟩ is continually clocked until UB INIT is deasserted.

The microsequencer goes to the first fork address, ⟨0F⟩ (hexadecimal), and remains there until breakout is produced by a UNIBUS or CMI-initiated transaction. Each microroutine returns to the first fork address upon completion.

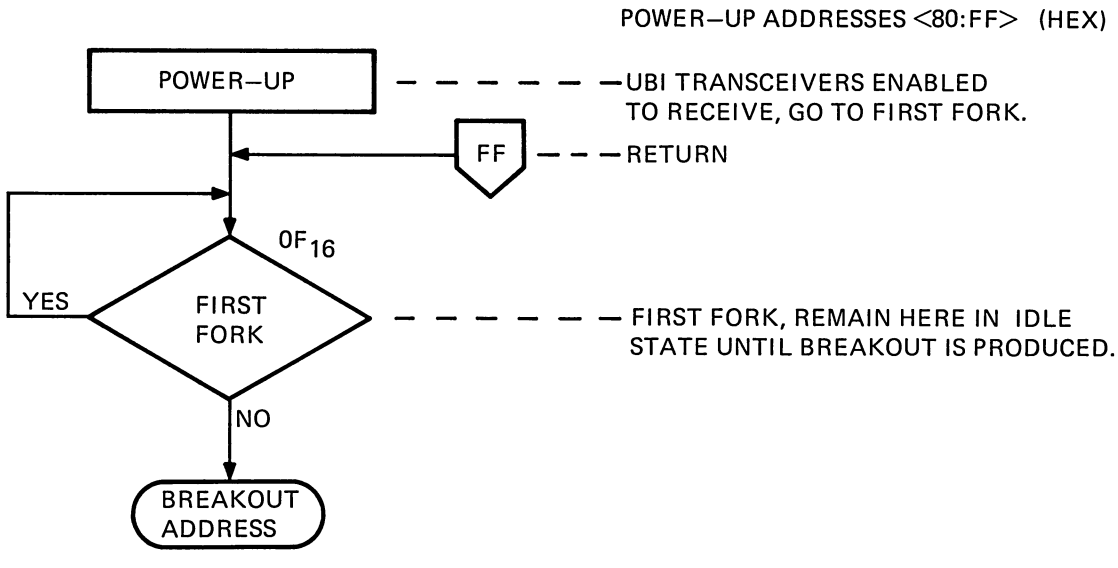


Figure 3-1 Power-Up Flow

3.1.2 UBI Microword

Figure 3-2 illustrates the UBI microword and bit fields that are presented in hexadecimal in the microprogram listing. The content of the microword in the illustration is the power-up code. The NEXT field specifies first fork address ⟨0F⟩ to be executed by the following B CLK L cycle.

The UCN directs microprogram sequences by monitoring conditions enabled by the BUT code and altering the NEXT field. (All fields are described in Section 3.4.) When released from the first fork state by one of fifteen breakout activities (Section 3.1.3), NEXT field bits ⟨22:16⟩ at the breakout address direct the microprogram to one of eight possible branch addresses depending on conditions tested by the BUT code.

Figure 3-2 is recommended for assistance in interpreting individual microword bits when referring to the listing. At breakout address 0A for example [DDP DATO(B)], the high-order bits read as DC (hexadecimal). Bit ⟨23⟩ (BUF CMI) is a 1 to assert the translated MAP outputs onto the BUF CMI lines to the UDP. The NEXT field actually points to 6C as the next location (the base address for the possible branches).

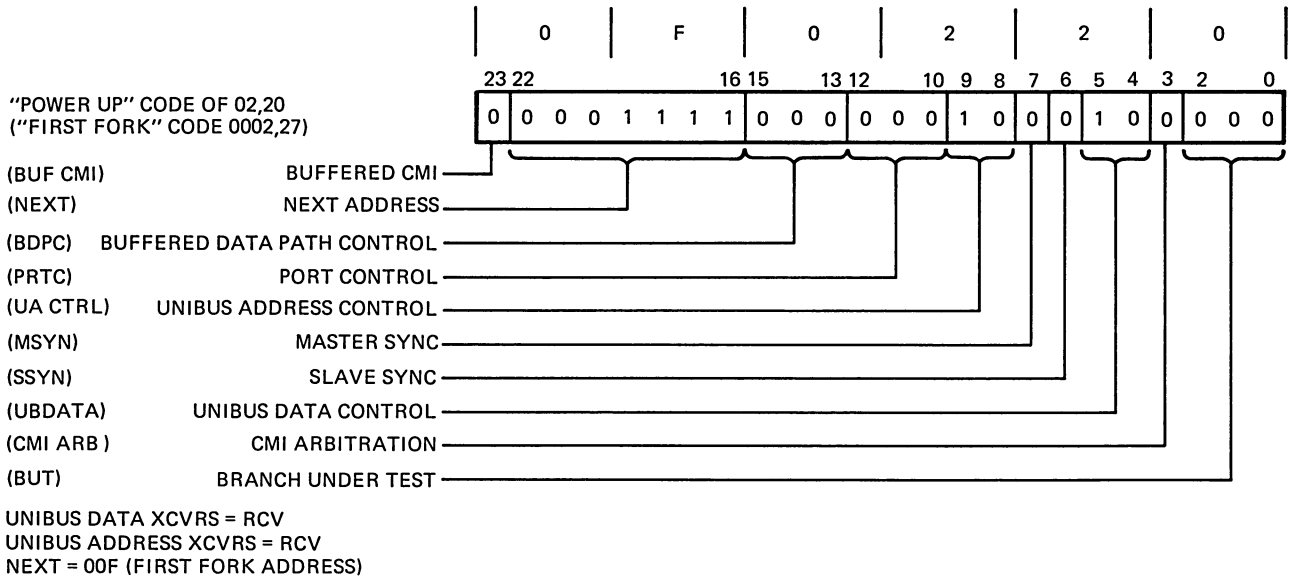


Figure 3-2 UBI Microword

3.1.3 First Fork Breakout

When the UCN detects breakout activity, it deasserts combinations of bits ⟨3:0⟩ of the 0F first fork address. This selects one of fifteen breakout addresses from ⟨00⟩ through ⟨0E⟩ (hexadecimal). The microinstruction at each breakout address initiates service for a specific UBI activity. Services initiated are listed in the following.

Table 3-1 Breakout Addresses

CMI Initiated Transactions

Breakout Address	Service
08	CPU write [DATO(B)] to the UNIBUS
09	CPU read [DATI(P)] from the UNIBUS

Table 3-1 Breakout Addresses (Cont)

UNIBUS Initiated Transactions	
Breakout Address	**Service**
03	BDP DATO and Not Full, UB data word is clocked to buffer.
01	BDP DATO and Full, UB data word is clocked to buffer and UBI write to CMI is generated.
03	BDP DATO and Wrap, lower UB data byte is clocked to buffer, UBI write to CMI is generated. Upper UB data byte is clocked to buffer at next longword address.
02	BDP DATOB and Not Full, UB data byte is clocked to buffer.
00	BDP DATOB and Full, UB data byte is clocked to buffer and UBI write to CMI is generated.
06	BDP DATOB and Wrap, upper UB data byte is clocked to buffer at next longword address.
07	BDP DATI, buffer Empty or No Match, device waits for UBI read on CMI and clocks data word from the received longword.
05	BDP DATI, Data Available and Match, device clocks data word from the longword in the buffer.
07	BDP DATI, buffer Empty and Wrap, device waits for two UBI read cycles on CMI to obtain a full word of data.
04	BDP DATI, Data Available and Wrap, device waits for UBI read on CMI to next longword address to complete the data word.
0A	DDP DATO(B), UBI write to CMI is generated, UB data word/byte is enabled to CMI data lines.
0A	DDP DATO-Wrap, UBI write to CMI is generated, lower UB data byte is enabled to CMI data lines. UBI write to CMI to next longword address is generated, upper UB data byte is enabled to CMI data lines.
0E	DDP DATOB-Wrap, UBI write to CMI to next longword address is generated, upper UB data byte is enabled to CMI data lines.
0B	DDP DATI(P), UBI read on CMI is generated, specified word of the CMI data longword is enabled to UB data latch, device clocks data word.
0B	DDP DATI and Wrap, UBI read on CMI is generated, one byte of CMI data is clocked to lower byte of UB data latch where it is held. UBI read on CMI is generated to next longword address,

Table 3-1 Breakout Addresses (Cont)

UNIBUS Initiated Transactions (Cont)	
Breakout Address	**Service**
0E	CMI data byte is clocked to upper byte of UB data latch, device clocks data word. BR interrupt (INTR), UBI performs write vector operation to CMI. Device vector address is asserted onto the CMI data lines.

Purge Operation	
Breakout Address	**Service**
0C/0D	Check byte flags for buffer empty on designated BDP. ● Byte flags set, generate UBI write to CMI (which clears byte flags). ● CD flag set, clear CD flag. ● No flags set, buffer Empty, no action.

3.1.4 UCN Defined BDP Transfer Conditions

The UCN determines microsequencer response for UNIBUS NPR transfers to the buffered data paths described in Section 2.6. The UCN defines the conditions that select the breakout addresses listed in Section 3.1.3.

The state of byte flags ⟨BF3:BF0⟩ determines the correct microsequence for UNIBUS DATO(B) data written to a BDP register (CD flag is clear):

Buffer Full — All byte flags are set. BDP buffer register contains four bytes of valid data for UBI transfer to main memory. Buffer full gating is an OR of the flags that are set and of the enable levels for the flags to be set by the current UNIBUS DATO(B) operation.

Not Full — One or more, but not all, byte flags are set. Buffer does not contain a full longword of valid data for a CMI transfer.

Empty — All byte flags are clear, buffer register contains no valid data.

Not Empty — Not all byte flags are clear, buffer contains valid data. Purge request or auto purge to the BDP generates UBI write to main memory.

The state of the CD flag determines the correct microsequence for UNIBUS DATI data on a BDP (byte flags are clear):

Empty — CD flag is clear. No DATI data is available in buffer. UBI read to main memory retrieves a longword of data.

Data Available — CD flag is set. Received CMI data is available in the buffer and can be read by the device.

The match level driven by the UDP (described in Section 2.4) determines whether a microsequence is selected which makes access to main memory:

No Match — The address contained in the BAR for the selected BDP does not agree with that specified by the UNIBUS device. Access to main memory is required to write existing UNIBUS DATO(B) data or to retrieve a longword of DATI data.

Match — The address contained in the BAR agrees with that specified by the UNIBUS device. The data word to be transferred is within the same longword address as the previous word.

Transfers to the same longword address do not generate UBI access to memory. Data is clocked only between the BDP register and the device. This allows two UNIBUS word transfers to take place before a single CMI longword transfer with main memory is required.

For this reason UNIBUS devices may not use a BDP location as a flag, monitoring it for changes by the CPU; nor can locked transfers to a BDP be supported. Locked transfers (DATIP) by a UNIBUS device are legal only on the direct data path with the MAP Offset bit cleared. A wrap condition which results with the offset bit set yields unpredictable results.

The wrap condition for UNIBUS DATO(B) transfers is introduced in Table 2-9 of Section 2.6.2.1. Details of UBI process of DATO(B) and DATI wrap conditions are provided in Sections 3.2 and 3.3.

Wrap — With MAP offset enabled, and UNIBUS byte 0, can be transferred with byte 3 of the CMI data longword at the current CM1 address. Reference is made to the next CMI address to transfer UNIBUS byte 1 with CMI data byte 0.

3.1.5 CMI and UNIBUS Protocol

The following diagrams and flowcharts illustrate UBI microprocessor response to transactions initiated by the CMI or by the UNIBUS, and how the signals are generated between them.

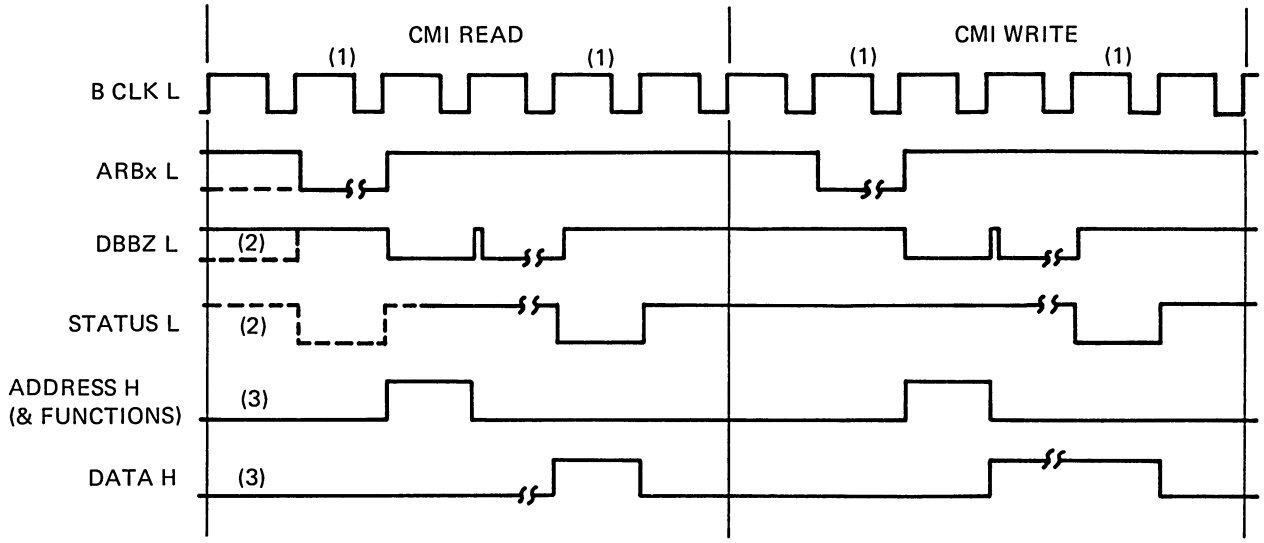
3.1.5.1 CMI Read/Write Cycles

Figure 3-3 is a timing diagram of read and write operations on the CMI. A minimum of three B CLK cycles is normally required to transfer one longword of data. These cycles are as follows.

1. Arbitration cycle (DBBZ not asserted).
2. CMI address cycle, CMI address and DBBZ asserted by master.
3. CMI data cycle, DBBZ asserted by slave.
   - Read cycle, slave deasserts DBBZ and returns data and status.
   - Write cycle, slave clocks data, deasserts DBBZ and returns status.

Actual time required for a transfer varies with the ability of a slave subsystem to return data or status. If a slave is immediately ready to receive write data, it does not assert DBBZ and only two cycles are required.

A subsystem may assert its arbitration level at any time. Arbitration takes place when DBBZ is not asserted. The subsystem with the highest priority arbitration level asserted holds off lower-priority subsystems. On the next positive transition of B CLK L, the new master asserts the physical longword



NOTES:
(1) ARBITRATION TAKES PLACE
(2) ASSERTED BY PREVIOUS TRANSACTION
(3) ASSERTED ON CMI DATA LINES

TK-5093

Figure 3-3 CMI Read/Write Cycles

address of the slave along with DBBZ. As described in Section 2.6.1.1, CMI Address Cycle, all other subsystems recognize that an address longword is present on the CMI and the addressed slave responds by asserting DBBZ.

3.1.5.2 UNIBUS NPR Cycles – Figure 3-4 is a general timing diagram of a UNIBUS NPR cycle. The arbitration cycle on the UNIBUS is a separate operation from the data transfer or write vector operations.

3.2 CMI ACCESS TO THE UNIBUS

Figures 3-5 and 3-6 illustrate UBI microprogram flow for a CPU write and CPU read to the UNIBUS. For the processor to access the UNIBUS:

1. The MIC module decodes the address to be sent to the CMI. For a UNIBUS address, MIC transmits UB REQ to the UBI on the backplane.

2. The UNIBUS is free, determined by the deasserted state of BBSY.

3. The arbitrator raises BBSY and holds it asserted as the CPU arbitrates for the CMI.

4. CMI address is transmitted by the CPU. The address field is clocked to the RCAR (UDP Section 2.4), and the UCN decodes the byte mask and function code to UNIBUS control bits. The ADDU level from the UDP is clocked to the latch ADDU flop in the UCN. This asserts DBBZ onto the CMI on the following bus clock cycle and selects the CPU read or write breakout address.

5. RCAR contents are gated to the UB address lines from the BUF CMI port of the UDP. (CMI data is clocked to the UB data latch for a CPU write.) Address (and data for a write) is asserted on the UNIBUS.

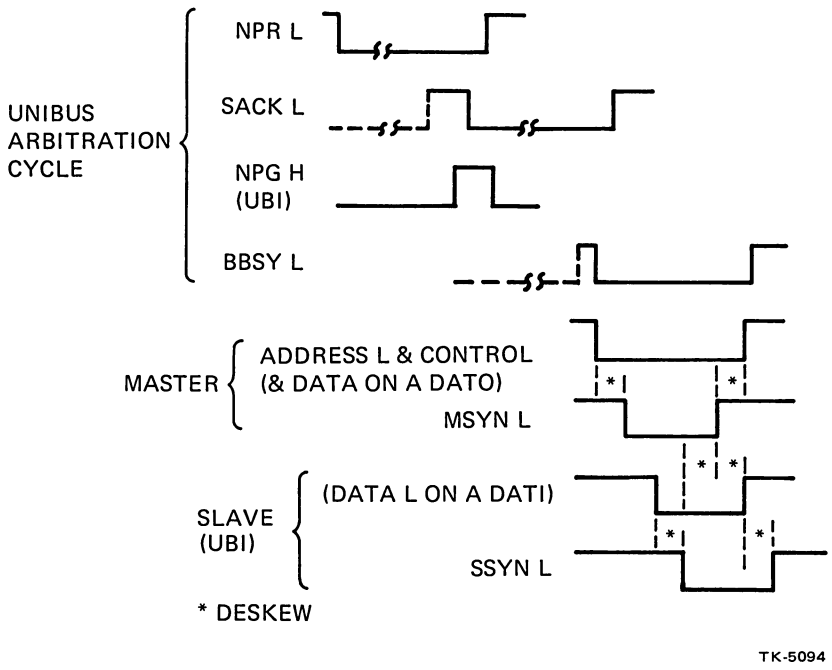


Figure 3-4 UNIBUS NPR Cycle

6. MSYN and HOLD are asserted by the UBI and the UNIBUS slave device responds with SSYN as described by the flowcharts.

Section 2.6.1 describes the details of control functions provided by the UCN. Figures 3-7 and 3-8 illustrate gating of the data through the UDP as directed by the BDPC and PRTC fields of the UBI microword during execution of the microprogram.

3.2.1 CPU Write (DATO/B)

Figure 3-7 illustrates UDP gating of CMI data to the UNIBUS. The lower and upper words of the CMI data longword are both enabled to UB data latch inputs from the byte select mux. For a DATO to the UNIBUS, the word clocked to the UB data latch is reflected by UB address bit ⟨A1⟩, decoded by the UCN from the CPU byte mask, and enabled to the UB address lines.

For a DATOB transfer, both bytes of the data word are enabled to the UB data lines. The upper or lower bytes (of the upper or lower word) to be clocked by the slave are specified by the decoded state of UB address bit ⟨A0⟩.

A CMI data byte is clocked by the device at the UB address that corresponds to the position of the byte in the CMI data longword. CMI data byte 0 is clocked at UB address XXXX00 (octal); CMI data byte 3 is clocked at UB address XXXX03, etc.

3.2.2 CPU Read (DATI)

Figure 3-8 illustrates UDP gating of CMI data from the UNIBUS. The UNIBUS data word is enabled to both upper and lower words of the CMI data longword and transmitted on the CMI data lines. The CPU clocks CMI data bytes as a function of the instruction it executes. The byte mask is decoded by the UCN to select bit ⟨A1⟩ of the UNIBUS device address.

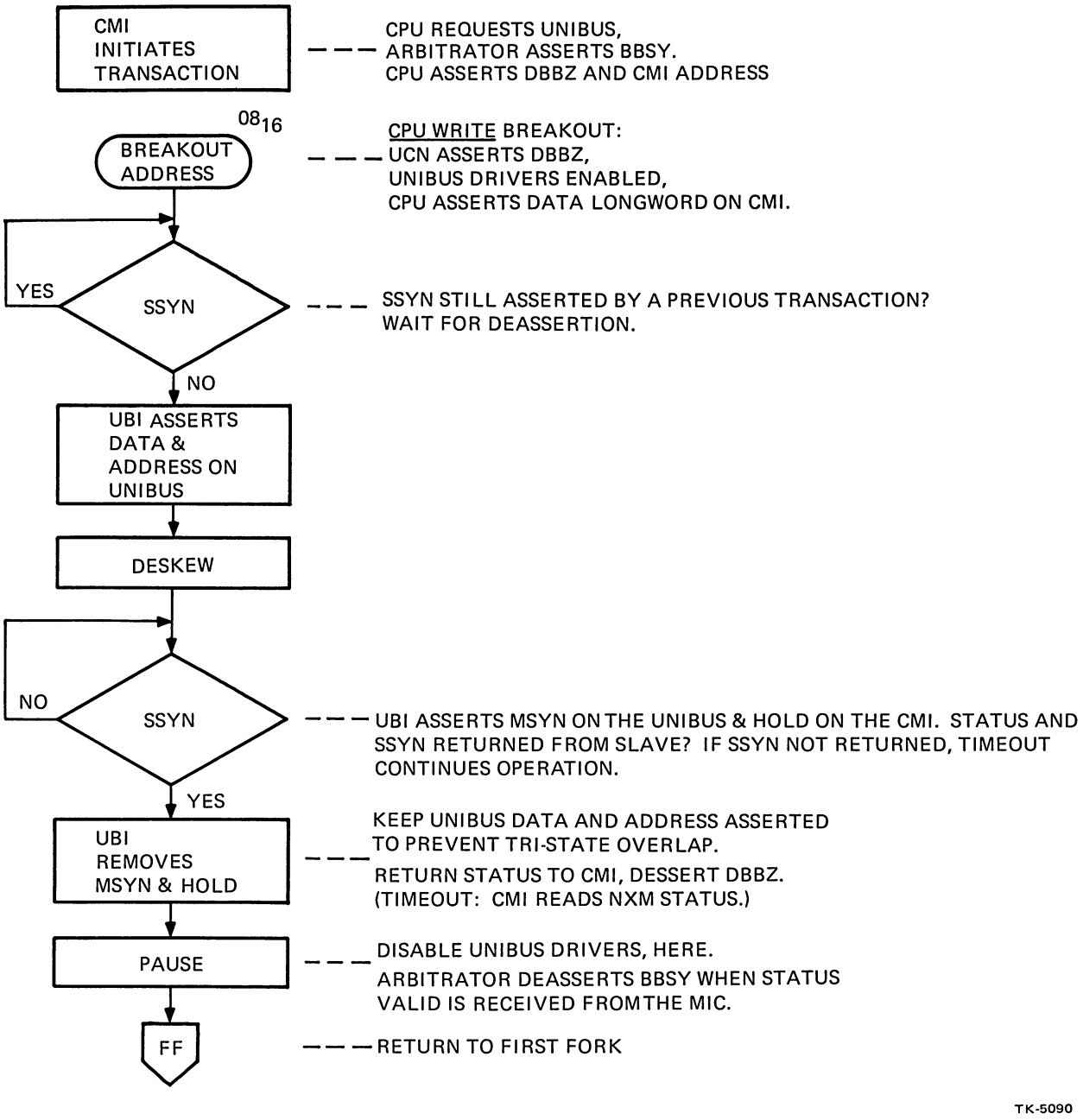


Figure 3-5 CMI Write to UNIBUS Flow, Breakout Address ⟨08⟩

3.2.3 CPU Read-Modify-Write (DATIP)

The CPU initiates a DATIP function with the UNIBUS by performing a locked read transfer on the CMI. BBSY is held asserted by the arbitrator (upon completion of the DATI) until the write unlock [DATO(B)] is performed.

3.3 UNIBUS ACCESS TO THE CMI

This section describes UBI coordination of UNIBUS-initiated mapped transfers to the CMI for transactions to main memory physical address space.

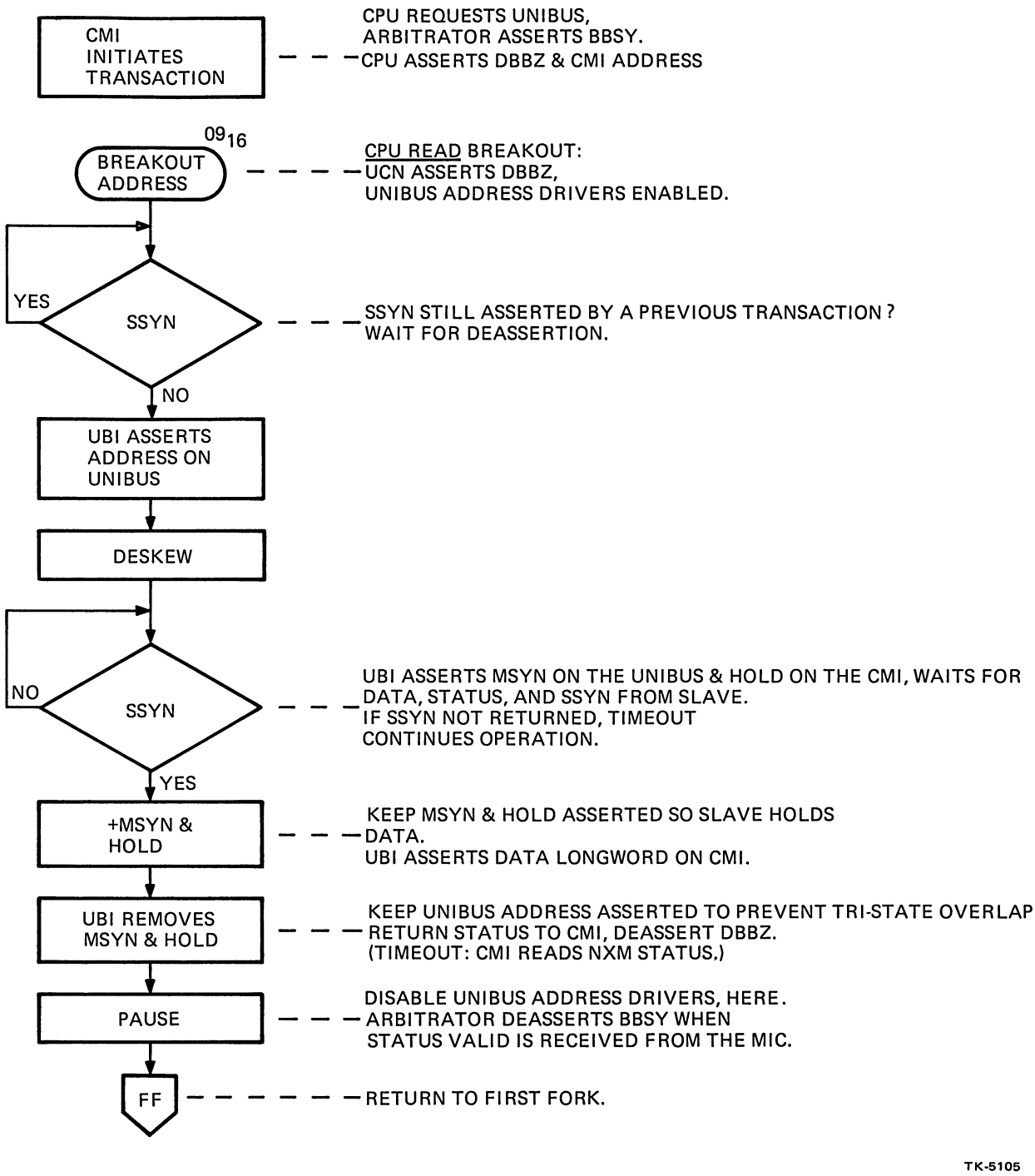


Figure 3-6 CMI Read from UNIBUS Flow, Breakout Address ⟨09⟩

Flowcharts with examples of data transfers illustrate clocking and gating of data within the UDP. References to UB data latch contents reflect DATI data enabled to the UNIBUS data lines. The CMI data longword reflects the contents of a BDP buffer register or data transferred through the CMI data port. Refer to Section 2.6.2 for the UCN-decoded, UBI-generated byte mask bits.

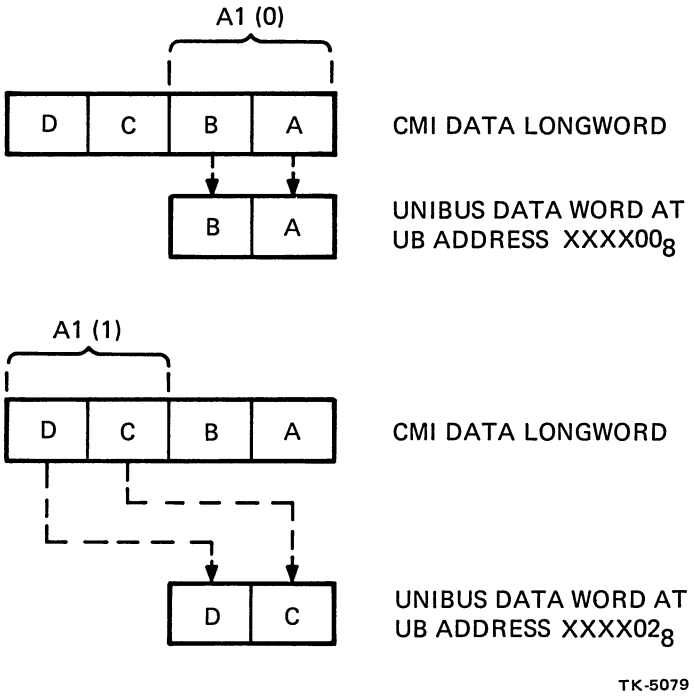


Figure 3-7 UBI Word Transfer to the UNIBUS

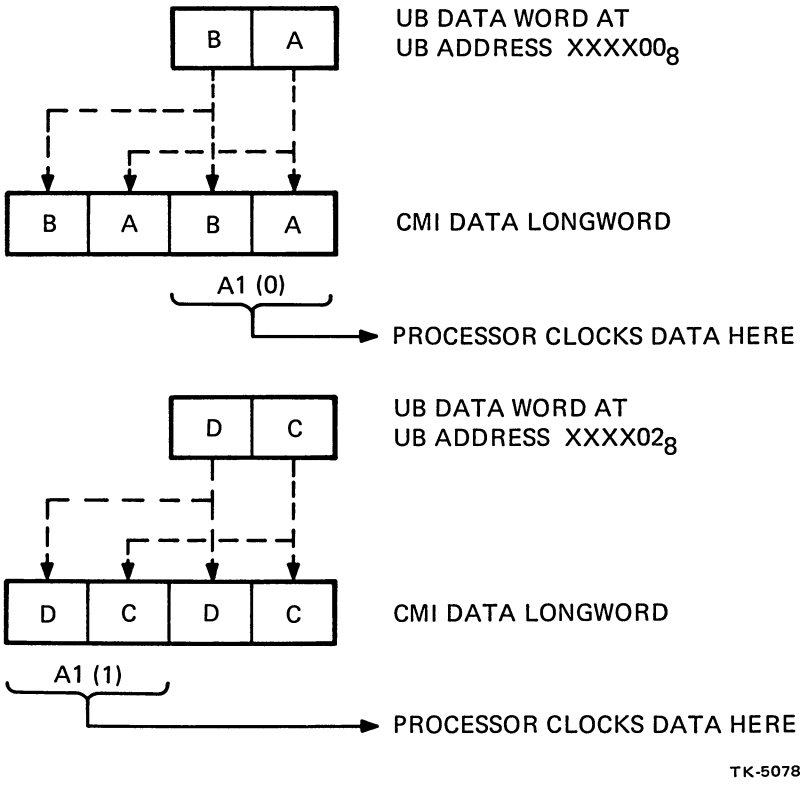


Figure 3-8 UBI Word Transfer from the UNIBUS

3.3.1 UNIBUS NPR Arbitration Cycle

Figure 3-9 illustrates the arbitration cycle as events occur on the UNIBUS in preparation for an NPR transaction with the UBI. The arbitration cycle is a separate bus cycle from the read or write transfer. For the UNIBUS to gain access to the CMI, the device raises its NPR level. A grant is issued to the requesting device which becomes the new master. The device asserts BBSY and UNIBUS address. The MAP address translation, with valid and offset bits and the selected data path, is available at this time and the breakout address is determined. Breakout occurs when the device asserts MSYN.

NOTE

This is the sequence of events that take place between a device on the UNIBUS and the arbitrator section of the UBI before breakout occurs to the UBI microcode (see Figure 3-4).

3.3.2 Direct Data Path Transfers

Figures 3-10 and 3-11 are microprogram flow of UNIBUS DATO(B) and DATI transfers on the DDP.

3.3.2.1 No Offset

DDP DATI – The NPR DATI from a UNIBUS device initiates a UBI read on the CMI and a longword of data is returned from main memory as in Figure 3-7. The lower or upper word of CMI data is clocked to the UB data latch, as directed by UB address bit ⟨A1⟩. UB data latch contents are asserted onto the UNIBUS data lines and SSYN is returned to the device.

DDP DATO(B) – The NPR DATO(B) from a UNIBUS device generates a UBI write on the CMI. As in Figure 3-8, the word received on the UB data lines is transmitted on both the upper and lower words of the CMI data lines. Main memory clocks data according to UBI byte mask bits, decoded by the UCN from the state of UB address bit ⟨A1⟩ for a DATO, and also of bit ⟨A0⟩ for a DATOB.

3.3.2.2 Offset

No Wrap – Figure 3-12 illustrates UDP gating with the MAP offset bit set to a 1 and UB address bit ⟨A1⟩ on a 0. A word of data is enabled between the UNIBUS data word and the upper and lower words of the CMI data longword, as in Figures 3-7 and 3-8. However, byte-swap/byte-select gating is enabled which swaps the position of the bytes within the data word as it is clocked to or from the UB data lines.

DDP DATO(B) – The swapped bytes of the UB data word are enabled to the CMI data lines, both upper and lower words, as they are for no offset. The byte mask generated by the UCN designates bytes ⟨2:1⟩ of the CMI data longword as valid data. The UB data word is effectively rotated one byte position to the left as it is clocked by main memory.

DDP DATI – Both upper and lower words of the CMI data longword from main memory are enabled to the input gating of the UB data latch. Bytes ⟨2:1⟩ of the CMI data longword are clocked to the UB data latch by the UCN-directed microprogram. The UB data word from main memory is effectively rotated one byte position to the right as it is clocked to the UB data latch and then read by the device.

Wrap to Next Longword – Figure 3-13 illustrates the effects of the MAP offset bit enabled, with UB address bit ⟨A1⟩ on a 1. The byte positions are swapped within the UB data word, as in Figure 3-12. However, two CMI transfers are required to transfer a word of data between the UB data lines and main memory.

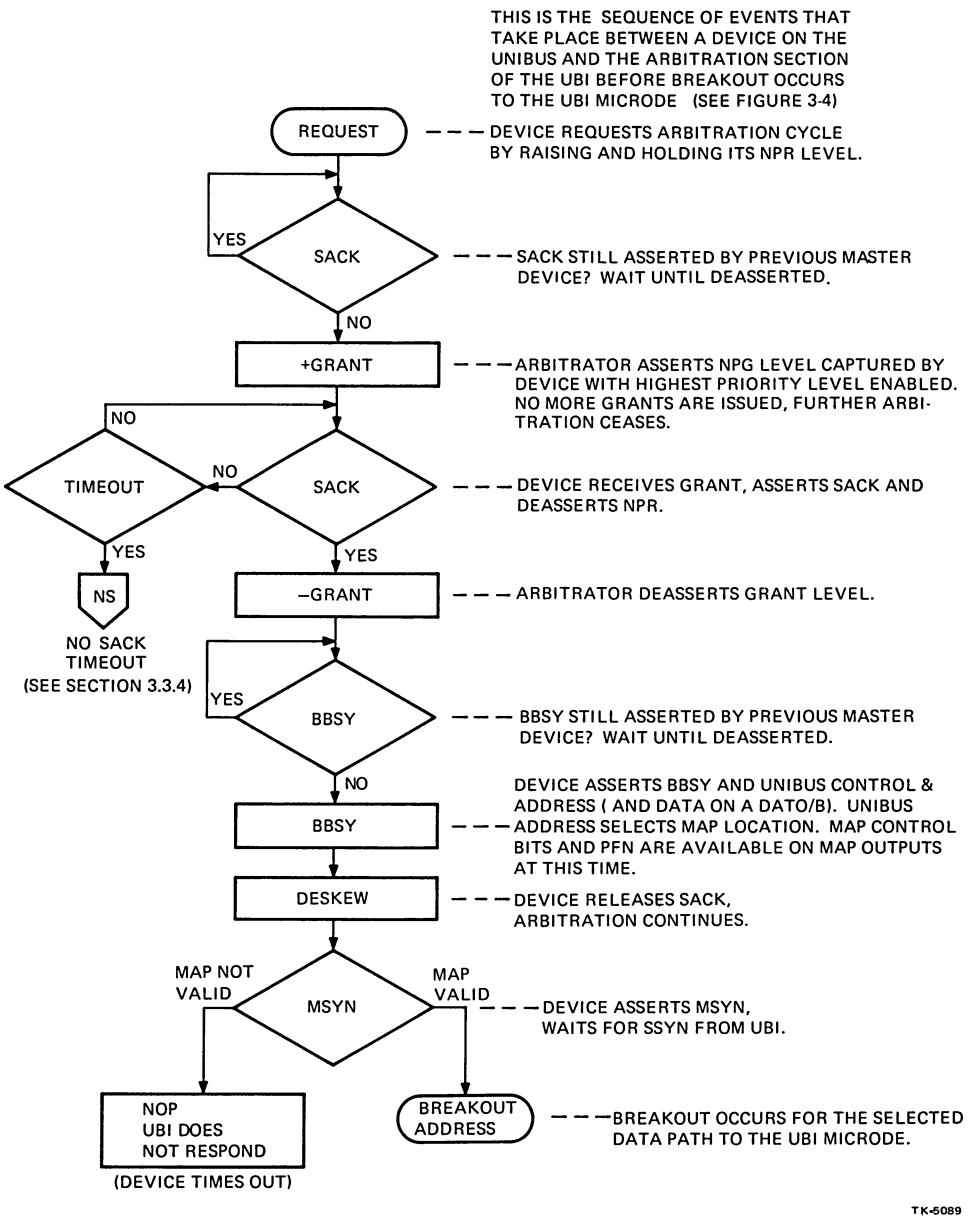


Figure 3-9 UNIBUS NPR Arbitration Flow

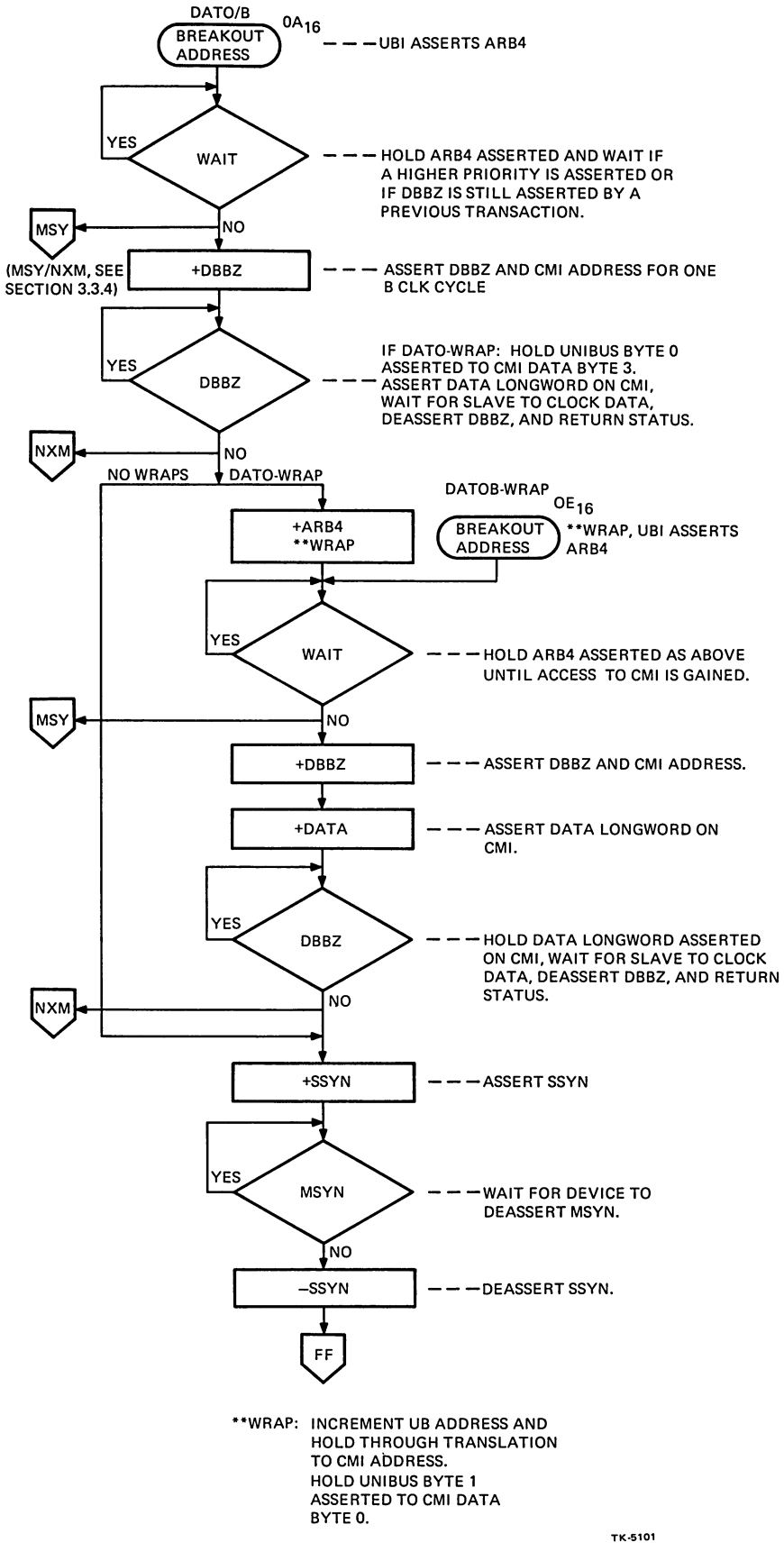


Figure 3-10 DDP DATO(B) Flow, Breakout Address ⟨0A,0E⟩

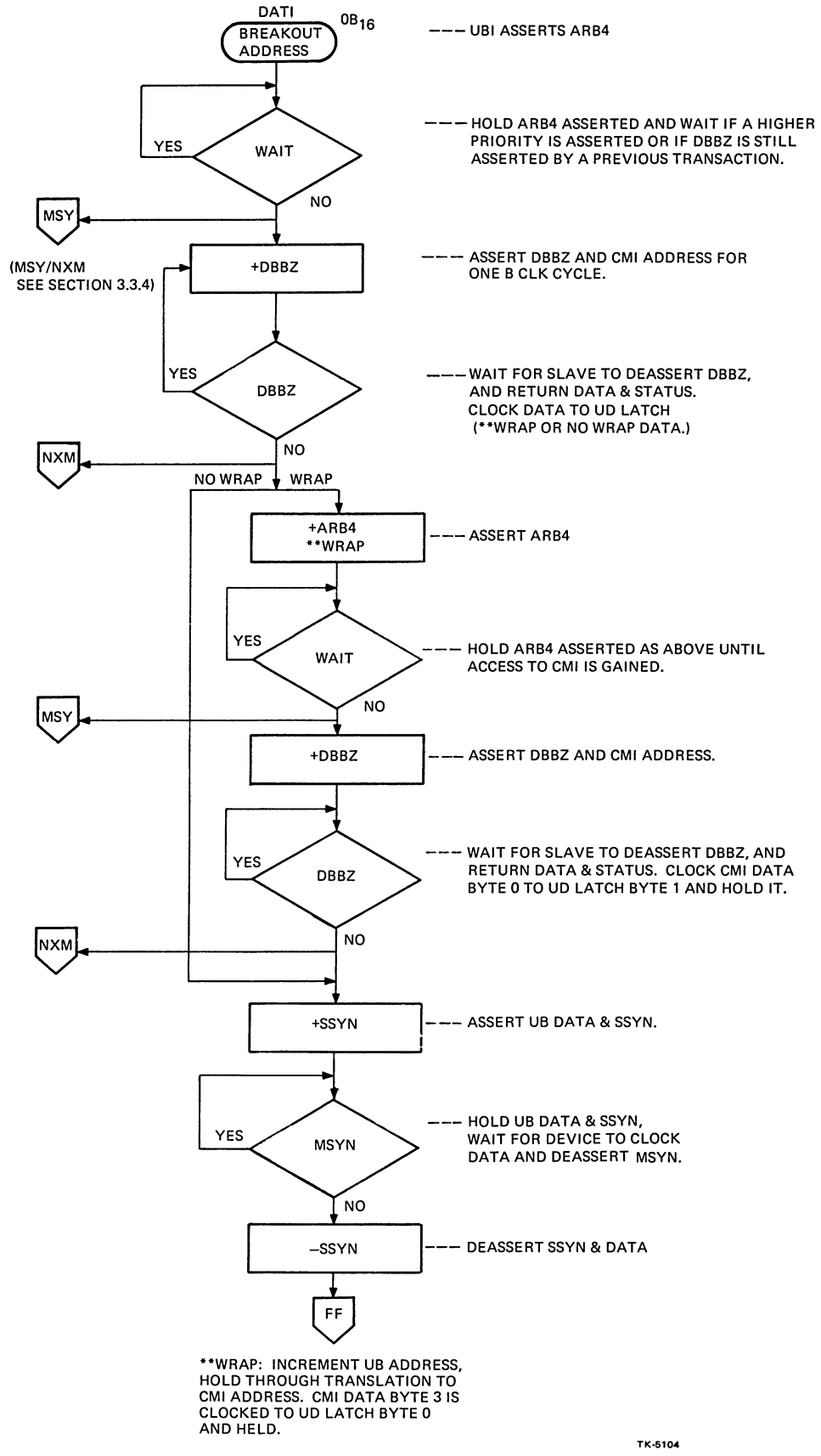


Figure 3-11 DDP DATI Flow, Breakout Address ⟨0B⟩

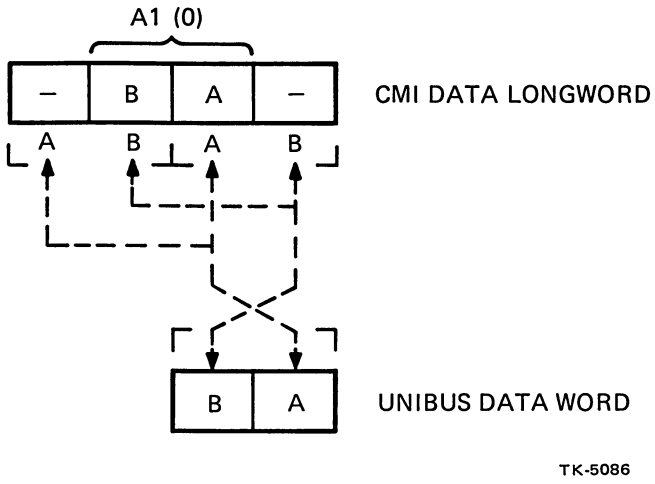


Figure 3-12 MAP Offset Enabled

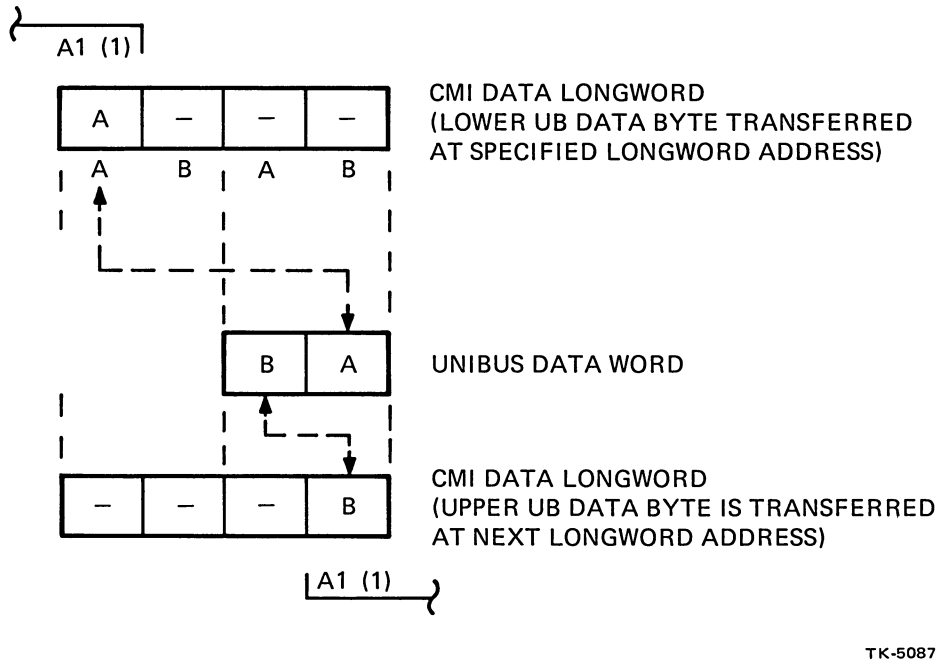


Figure 3-13 MAP Offset and Wrap

DDP DATI and Wrap – A UBI read to main memory retrieves a longword of data, and byte 3 of the CMI data longword is clocked to byte 0 of the UB data latch where it is held. Another UBI read to main memory retrieves a longword of data at the next longword address. Byte 0 of the CMI data longword is clocked to byte 1 of the UB data latch. SSYN is returned to the device which then clocks the data word from the UB data latch.

DDP DATO(B) and Wrap – The NPR from the device generates a UBI write on the CMI and the byte mask designates byte 3 as valid data for main memory. Another UBI write is generated to the next longword address and the byte mask designates byte 0 as valid data. SSYN is then returned to the device.

For a DATOB offset with UB address bit ⟨A1⟩ on a 1, a single UBI read/write is generated to transfer UNIBUS data byte 0 at the specified longword address (no wrap), or byte 1 at the next longword address (wrap), as directed by UB address bit ⟨A0⟩.

3.3.3 Buffered Data Path Transfers

3.3.3.1 BDP DATO(B) – Figure 3-14 illustrates microprogram flow for UNIBUS DATO(B) transfers on a buffered data path. Figures 3-15 thru 3-18 illustrate how the UBI microprocess handles data for the transfer conditions defined in Sections 3.1.3 and 3.1.4.

Figure 3-15 provides two examples of UNIBUS DATO transfers to a BDP with MAP offset set to a 0. In the first example, UB address bit ⟨A1⟩ is set to a 0 for the starting address; it is set to a 1 in the second. In both cases, transfers are begun to an empty buffer on even starting addresses (bit ⟨A0⟩ on a 0). The contents of a BDP buffer register are shown with the state of the byte flags as a result of a transfer operation with the UNIBUS.

Figure 3-16 is an example of UNIBUS DATO transfers to a BDP with MAP offset set to a 1. The result of the enabled offset bit is to rotate the UNIBUS data word one byte position to the left as it is clocked to the BDP register.

The wrap condition is introduced:

- Lower UB data byte is clocked to the BDP register and a UBI write is generated to the CMI.
- Upper UB data byte is clocked to the BDP register, the UNIBUS address is incremented and clocked to the BAR register. This indicates to the next transfer to that BDP that the data word is in the same longword address. The match condition allows further data to be clocked to the BDP register which now contains the upper byte from the previous transfer.

Figure 3-17 describes UNIBUS DATOB transfers with MAP offset set to 0.

In the second example, auto purge is generated for an attempt by the UNIBUS to write data to a BDP register which contains data; but the UNIBUS address does not agree with the contents of the BAR register (no match). The UCN generates a UBI write to the CMI to transfer BDP register contents before continuing the UNIBUS transaction.

Figure 3-18 describes DATOB transfers with the MAP offset bit set to a 1. The data byte is rotated one byte position to the left as it is clocked to the BDP register. This has the effect of storing the byte at the UNIBUS-specified byte address plus one.

In the second example, at a starting UB address of XXXX03 (octal) with the offset bit set, the UNIBUS address is incremented and clocked to the BAR as the upper UB data byte is clocked to the BDP buffer register.

3.3.3.2 BDP DATI – Figure 3-19 illustrates microprogram flow for UNIBUS DATI transfers on a buffered data path. Figures 3-20 and 3-21 illustrate UBI handling of data for the transfer conditions defined in Sections 3.1.3 and 3.1.4.

Figure 3-20 describes UNIBUS DATI transfers from a BDP buffer register at both even starting addresses with the MAP offset bit cleared.

The first NPR generates a UBI read to the CMI to retrieve a longword of data from main memory and set the CD Flag. Data is clocked both to the buffer and to the UB data latch from the CMI latch. A subsequent read [with data available (CD flag set) and address match] clocks buffer data to the UB

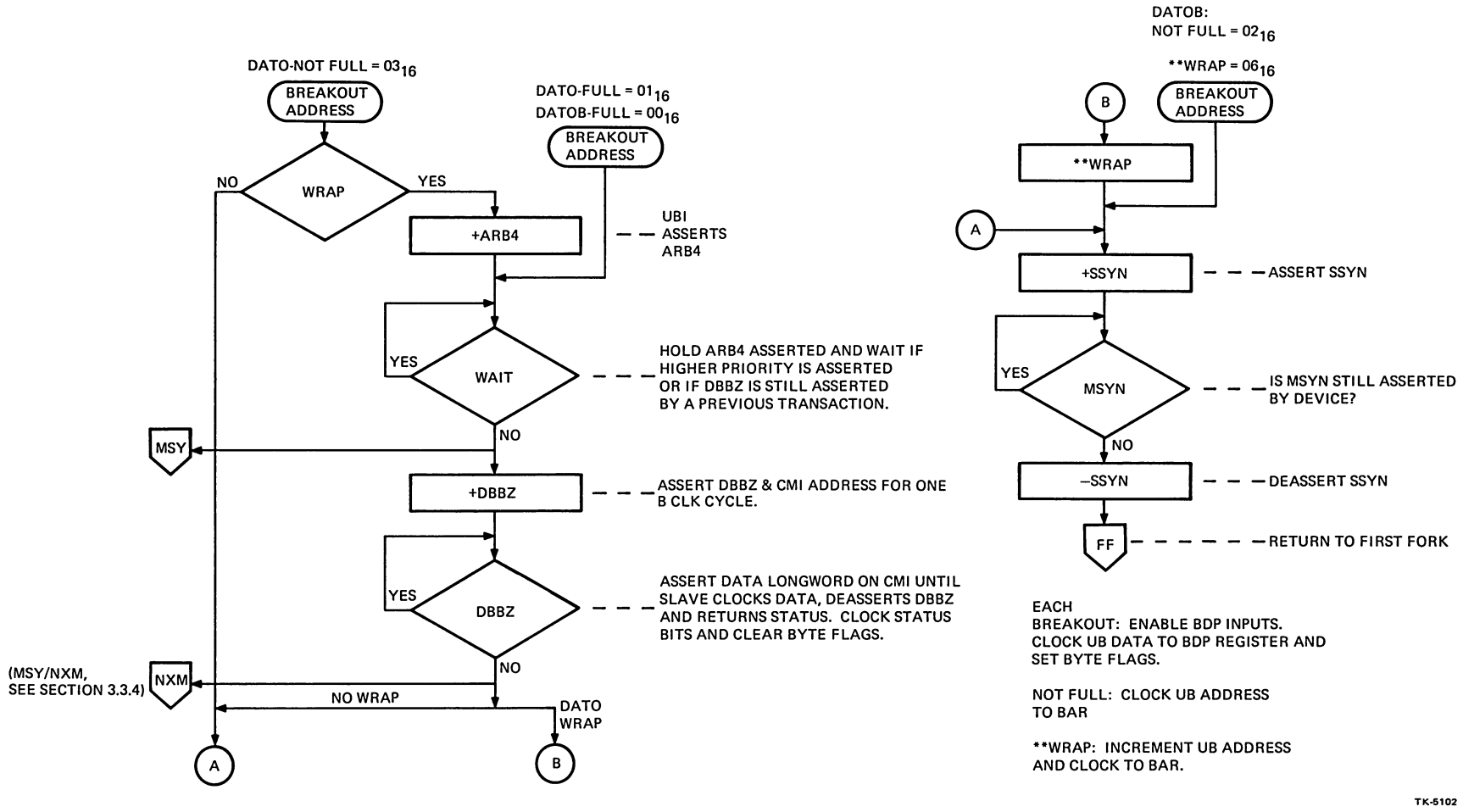


Figure 3-14 BDP DATO(B) Flow, Breakout Addresses ⟨06, 03:00⟩

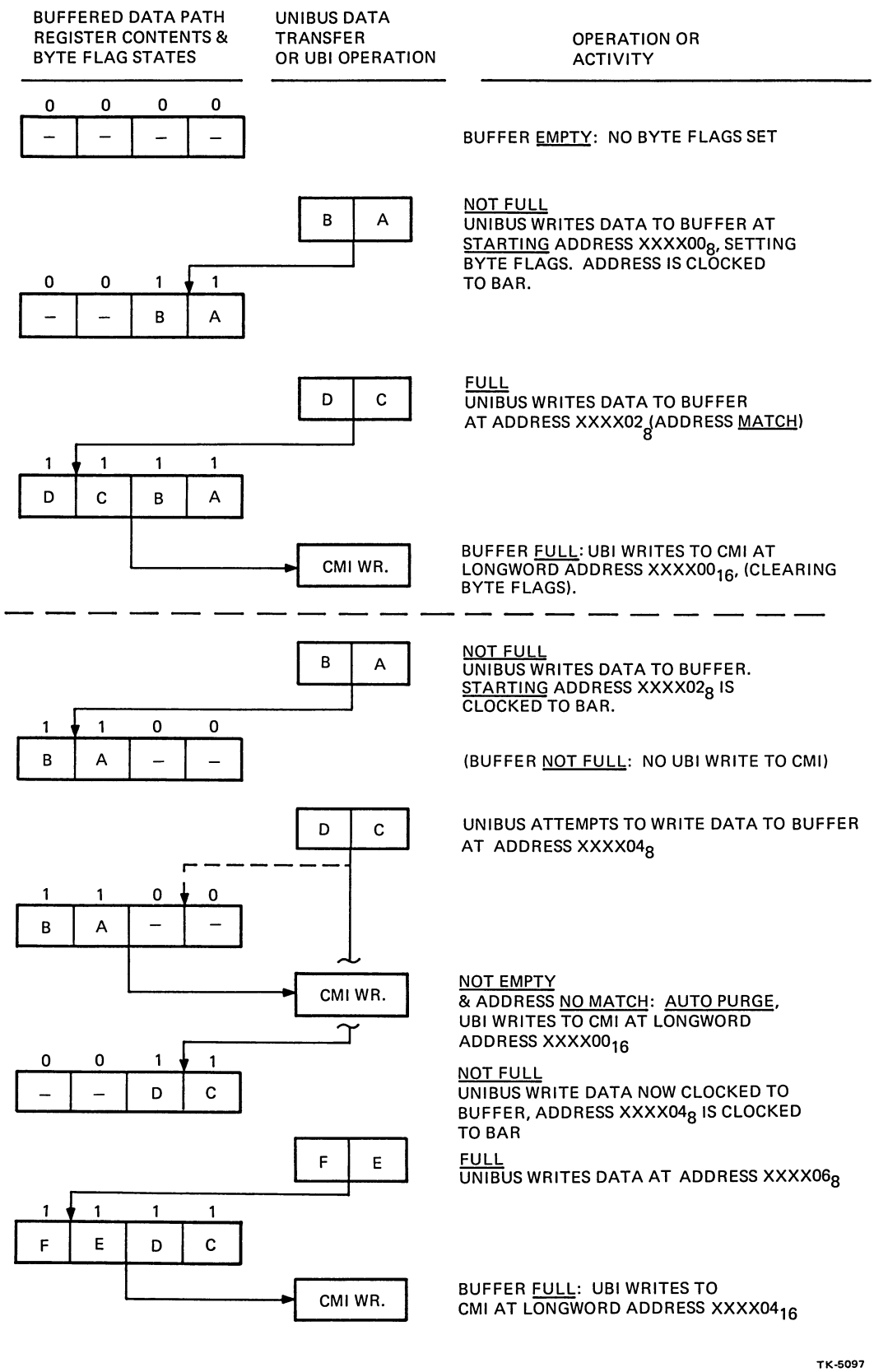


Figure 3-15 DATO on Buffered Data Path

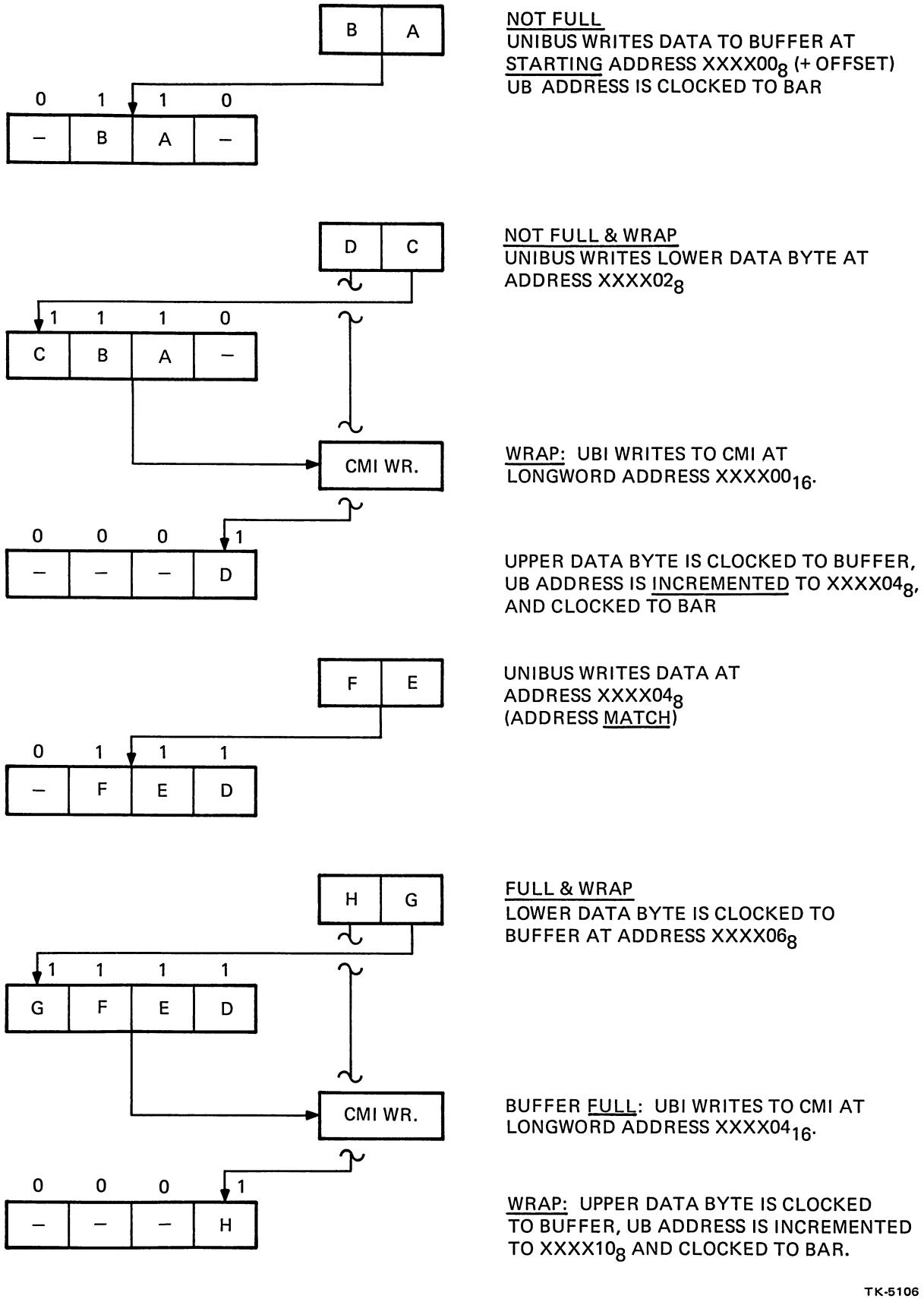


Figure 3-16 DATO and Offset on Buffered Data Path

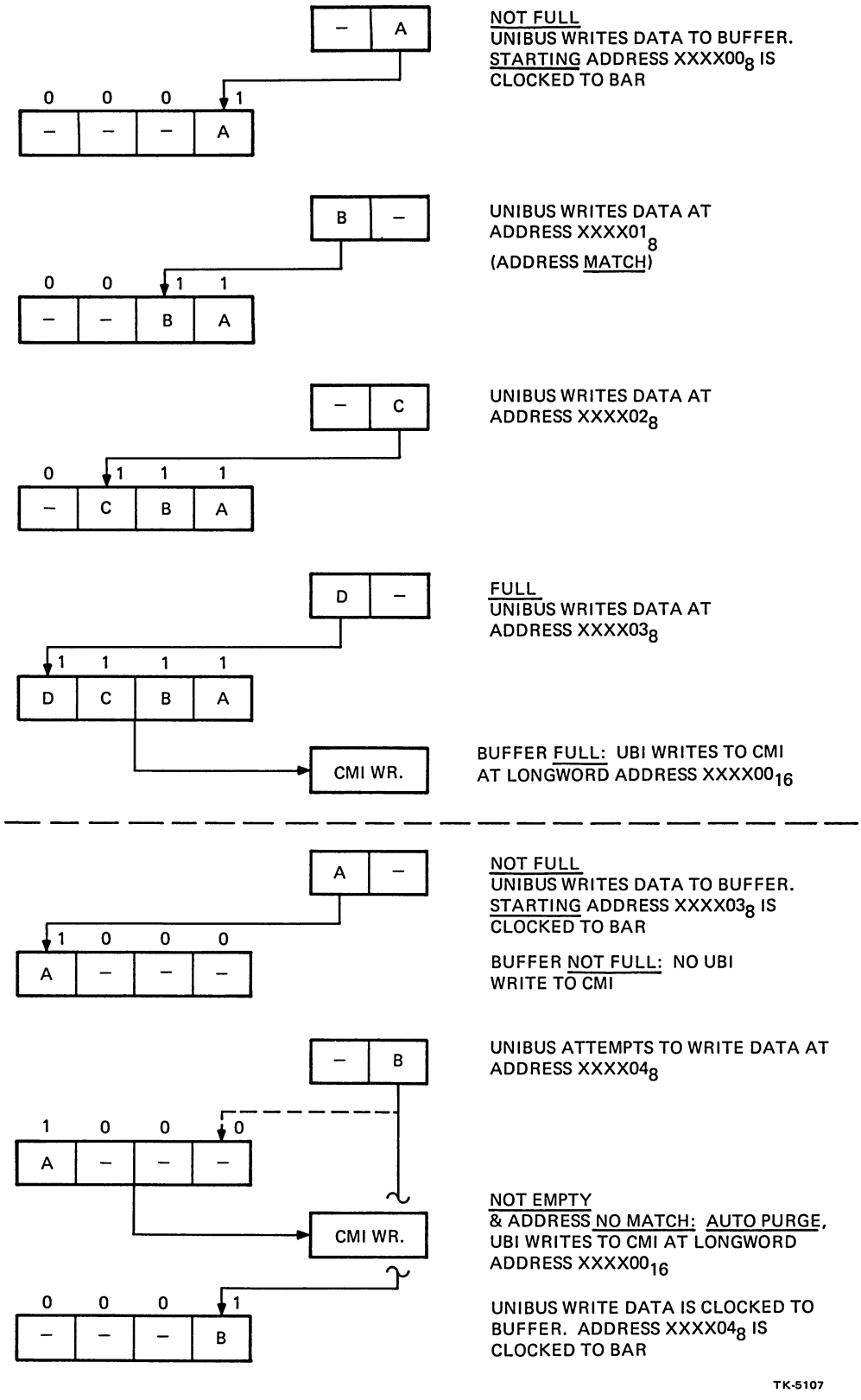


Figure 3-17 DATOB on Buffered Data Path

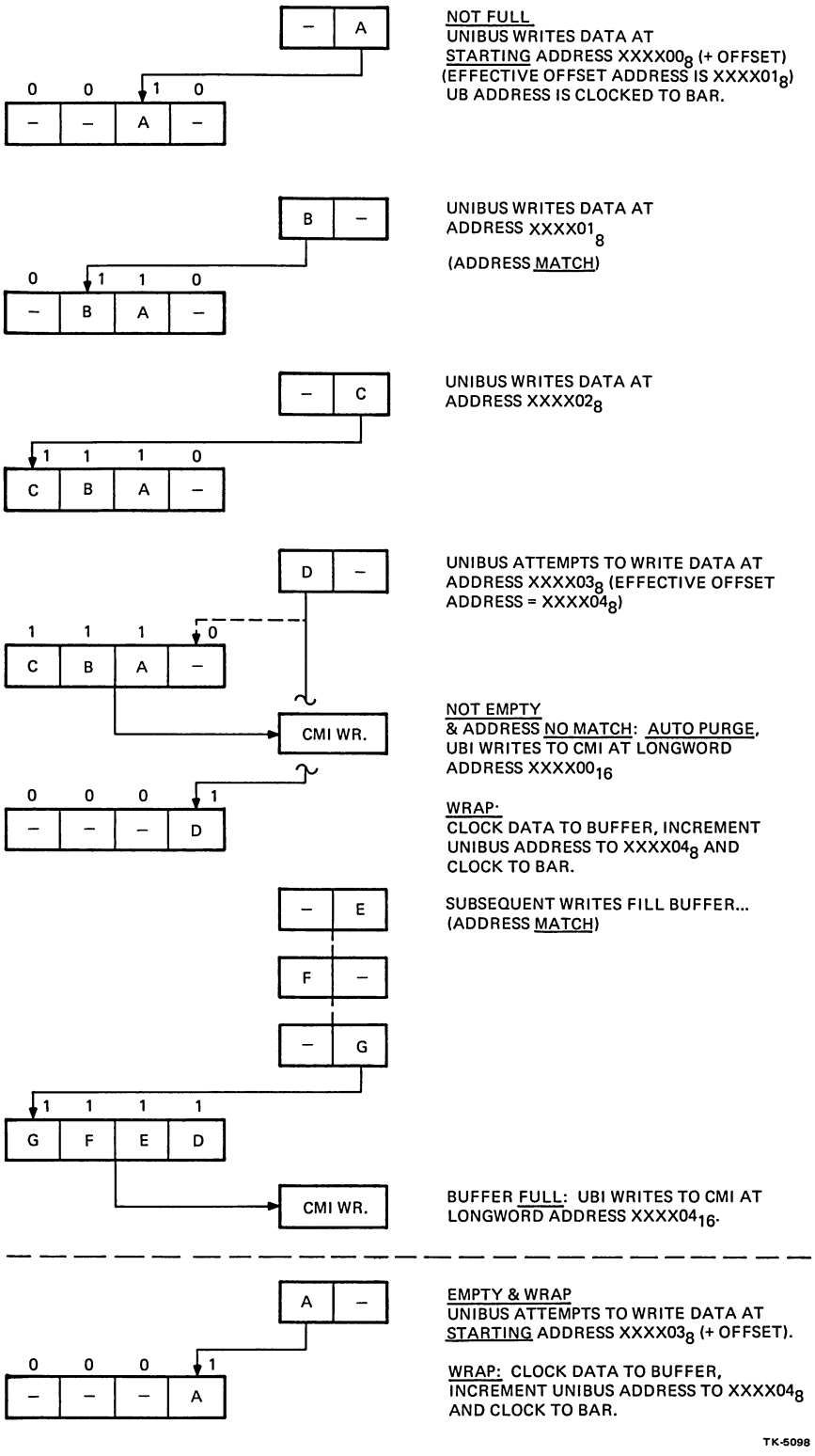


Figure 3-18 DATOB and Offset on Buffered Data Path

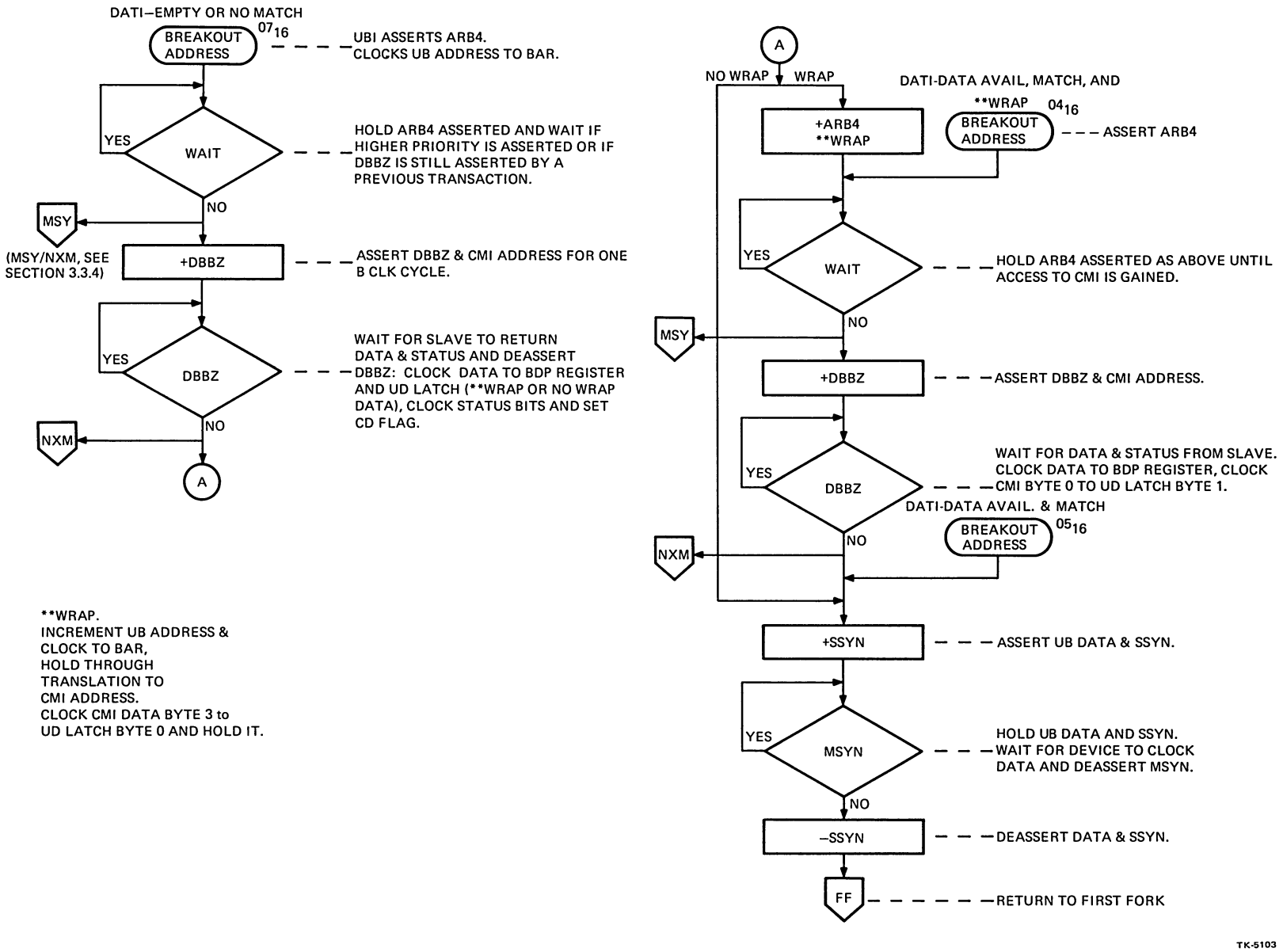


Figure 3-19 BDP DATI Flow, Breakout Addresses ⟨07,05,04⟩

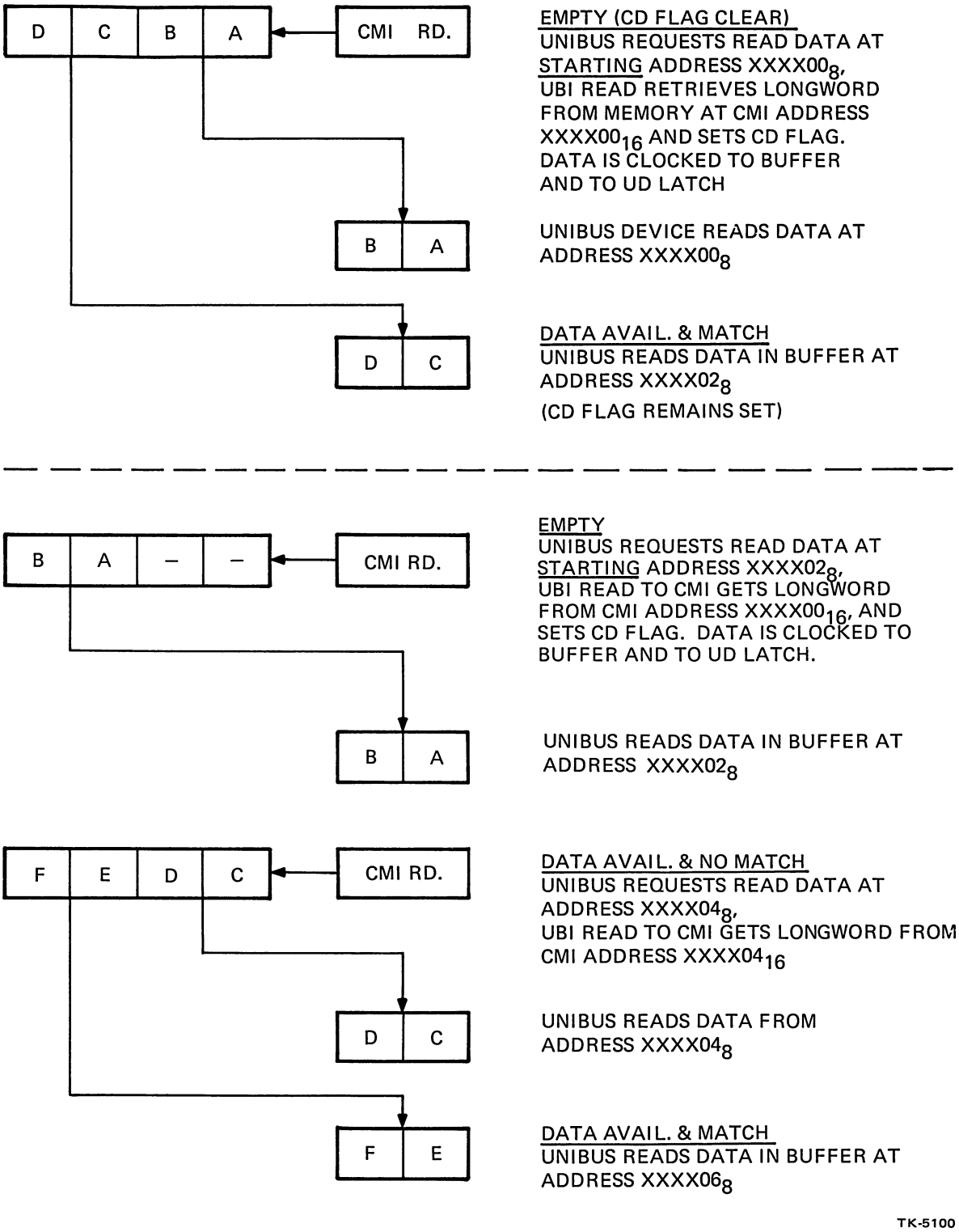
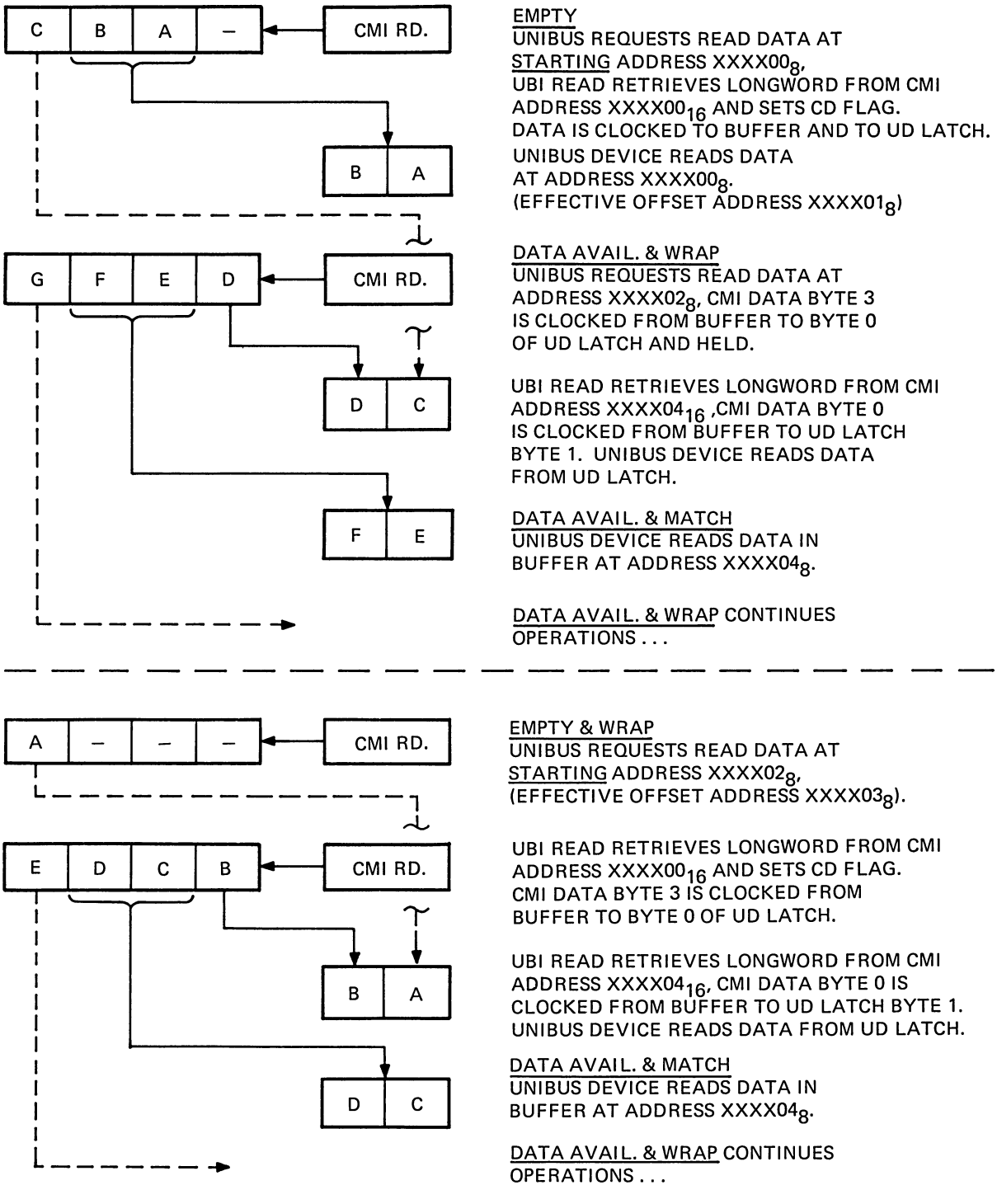


Figure 3-20 DATI on Buffered Data Path



EMPTY
UNIBUS REQUESTS READ DATA AT STARTING ADDRESS $XXXX00_8$,
UBI READ RETRIEVES LONGWORD FROM CMI ADDRESS $XXXX00_{16}$ AND SETS CD FLAG.
DATA IS CLOCKED TO BUFFER AND TO UD LATCH.
UNIBUS DEVICE READS DATA AT ADDRESS $XXXX00_8$.
(EFFECTIVE OFFSET ADDRESS $XXXX01_8$)

DATA AVAIL. & WRAP
UNIBUS REQUESTS READ DATA AT ADDRESS $XXXX02_8$, CMI DATA BYTE 3 IS CLOCKED FROM BUFFER TO BYTE 0 OF UD LATCH AND HELD.

UBI READ RETRIEVES LONGWORD FROM CMI ADDRESS $XXXX04_{16}$,CMI DATA BYTE 0 IS CLOCKED FROM BUFFER TO UD LATCH BYTE 1. UNIBUS DEVICE READS DATA FROM UD LATCH.

DATA AVAIL. & MATCH
UNIBUS DEVICE READS DATA IN BUFFER AT ADDRESS $XXXX04_8$.

DATA AVAIL. & WRAP CONTINUES OPERATIONS . . .

EMPTY & WRAP
UNIBUS REQUESTS READ DATA AT STARTING ADDRESS $XXXX02_8$,
(EFFECTIVE OFFSET ADDRESS $XXXX03_8$).

UBI READ RETRIEVES LONGWORD FROM CMI ADDRESS $XXXX00_{16}$ AND SETS CD FLAG.
CMI DATA BYTE 3 IS CLOCKED FROM BUFFER TO BYTE 0 OF UD LATCH.

UBI READ RETRIEVES LONGWORD FROM CMI ADDRESS $XXXX04_{16}$, CMI DATA BYTE 0 IS CLOCKED FROM BUFFER TO UD LATCH BYTE 1. UNIBUS DEVICE READS DATA FROM UD LATCH.

DATA AVAIL. & MATCH
UNIBUS DEVICE READS DATA IN BUFFER AT ADDRESS $XXXX04_8$.

DATA AVAIL. & WRAP CONTINUES OPERATIONS . . .

TK-5099

Figure 3-21 DATI and Offset on Buffered Data Path

data latch to be read by the device. In the second example, if an address match does not exist, a UBI read is generated to the next CMI longword address.

Figure 3-21 describes UNIBUS DATI transfers with the MAP offset bit set. For a starting UB address of XXXX00 (octal), a UBI read is generated to the CMI for the first data word. A wrap condition occurs on the next NPR. In the second example with a starting UB address of XXXX02, two UBI read operations take place on the CMI to construct the first data word to be clocked by the device.

3.3.3.3 BDP Purge – Figure 3-22 represents purge flow for an operation initiated to a BDP. The BDP is selected either by the software (purge request) or by the MAP as a result of a UNIBUS NPR (auto purge).

If any byte flags are set for UNIBUS DATO(B) data in the buffer (buffer not empty), a UBI write transfers the data to main memory. The byte flags are then cleared. If no byte flags are set (buffer empty), the other leg of the flowchart is executed. If the CD flag is set, it is cleared.

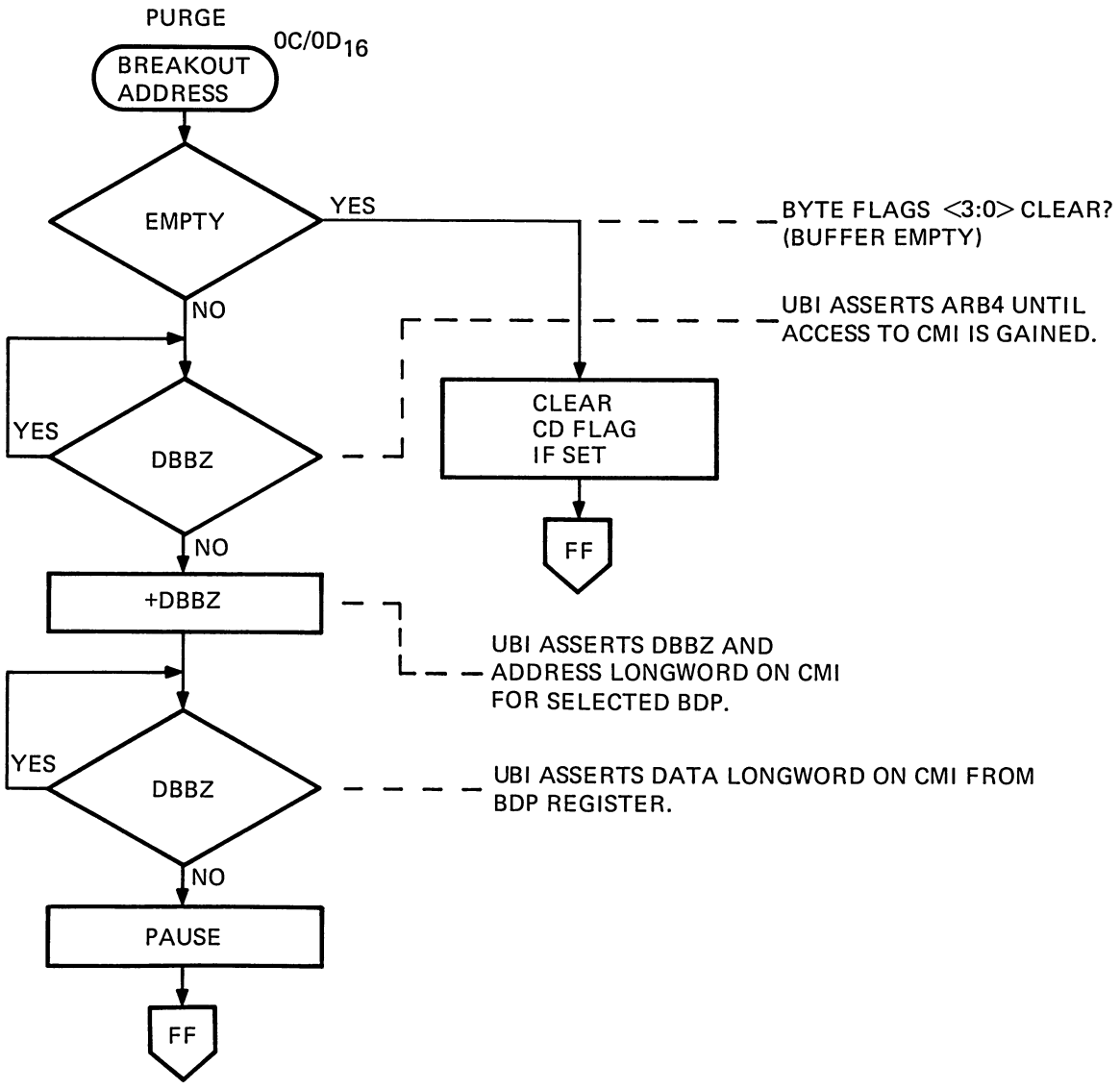


Figure 3-22 Purge Flow, Breakout Addresses ⟨0C,0D⟩

3.3.4 Error Flows

Figure 3-23 contains flowcharts for the nonexistent memory (NXM) condition, and for a device that has timed-out and deasserted the MSYN level (MSY).

The No SACK Timeout sequence is as follows.

1. UBI abitrator receives a bus request.
2. CPU microcode directs the UBI to issue bus grant and stalls the M CLK until the write vector occurs or SACK is deasserted.
3. Timeout counter asserts SACK.

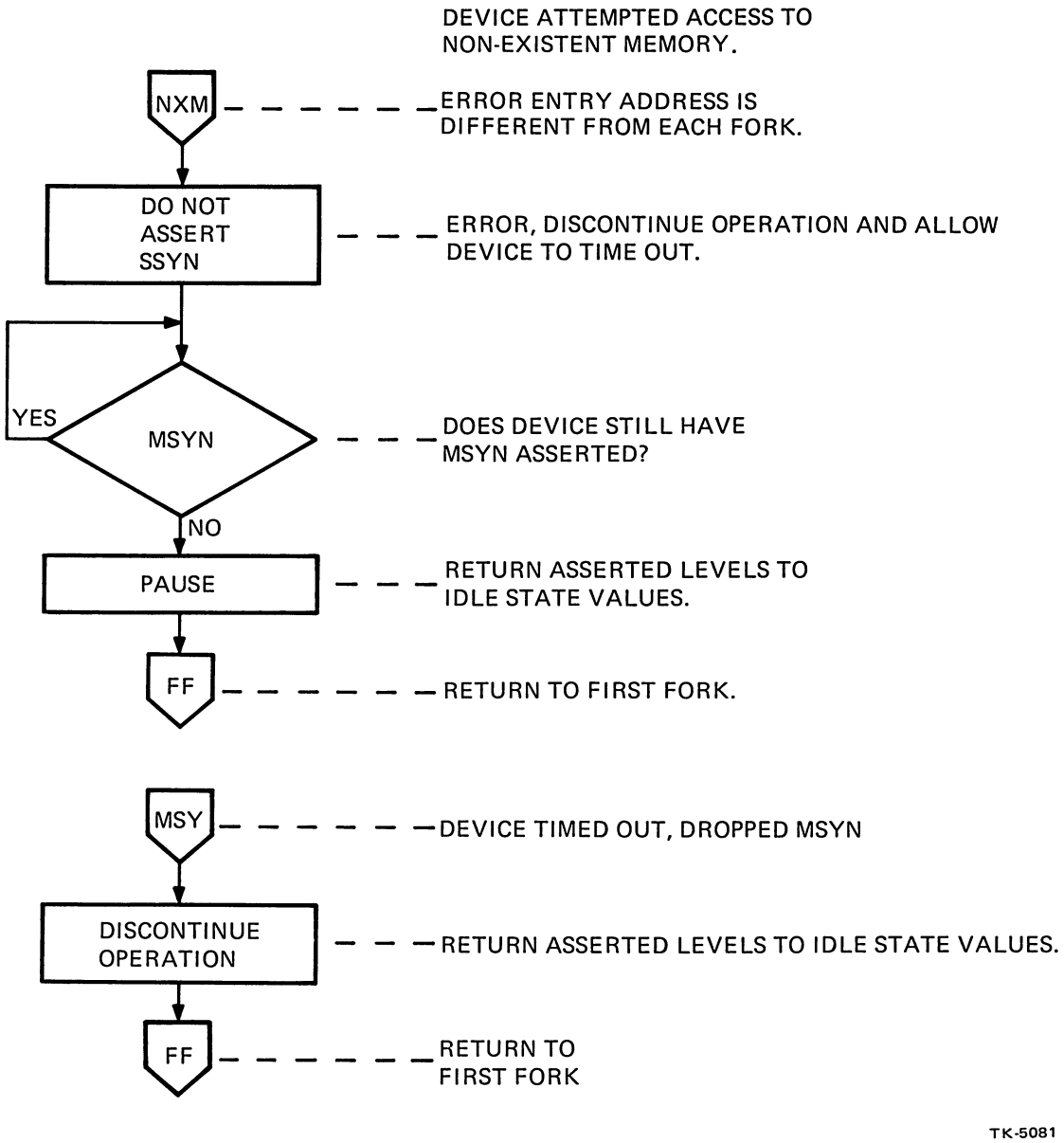


Figure 3-23 Error Flows

4. SACK deasserts bus grant.

5. Bus grant removal deasserts SACK which unstalls M CLK. The operation is considered complete.

6. CPU microcode examines the interrupt register to determine that INTR did not occur.

3.3.5 BR Interrupt/Write Vector

The arbitration cycle for a UNIBUS BR interrupt is similar to an NPR. The only exception to this similarity is in the dialog which takes place between the UNIBUS and the CPU before the processor can field the write vector operation. Figure 3-24 illustrates the UNIBUS BR arbitration flow; Figure 3-25 illustrates UBI write vector flow.

A BR priority level generated by a UNIBUS device is latched by the M CLK signal and asserted as the appropriate SBR level to the INT chip (Section 3.6). The INT chip compares the SBR ⟨7:4⟩ level which corresponds to an IPL ⟨17:14⟩ level. If the SBR is higher than the processor IPL:

1. INT PEND signal is updated at each trailing edge of M CLK and sent to the DPM and MIC modules.

2. INT chip selects MICRO VECTOR ⟨2:0⟩ lines to identify the type of interrupt pending. The value is 2 for a UNIBUS-originated interrupt.

INT PEND is used by the CPU to generate remaining MICRO VECTOR ⟨5:3⟩ lines to select the microvector address that services the incoming interrupt:

1. INT PEND is received by the SAC chip on the DPM while macrocode is running but is not interpreted until IRD1 of the next microinstruction.

2. The SAC chip generates the DO SERVICE and ENABLE uVECTOR signals to the MSQ chip which selects MICRO VECTOR ⟨5:4⟩ bits.

3. INT PEND to the uTRAP chip on the MIC selects MICRO VECTOR ⟨3⟩ bit.

Selected MICRO VECTOR bits ⟨5:3⟩ with bits ⟨2:0⟩ from the UBI direct the CCS to the interrupt handling microroutine. The first function of the microroutine is to send a 33 (hexadecimal) on the WCRTL ⟨5:0⟩ lines to the INT logic which enables the bus grant (BGn) level to be returned to the device. UB INT GRANT is also sent to the CMK chip on the MIC module. The CMK generates GRANT STALL which stalls the CPU microcode until the vector is written to the MIC module or WAIT is deasserted.

When SACK is returned by the requesting device, WAIT is asserted on the CMI. WAIT is received by the MIC module which replaces UB INT GRANT to hold the CPU stalled. When the device can assert BBSY and the vector address, it then asserts INTR which holds WAIT asserted on the CMI to maintain the CPU stall; breakout then occurs to the UBI microroutine for a DDP DATA. The INTR level directs the microprogram to a different branch which performs a write vector operation on the CMI.

Figures 3-26 and 3-27 are general timing diagrams of the activities that take place on the UNIBUS and the CMI for a write vector operation. A minimum of two bus clock cycles is required for a write vector on the CMI. DBBZ is not asserted by the CPU, the vector address is clocked directly and status is returned.

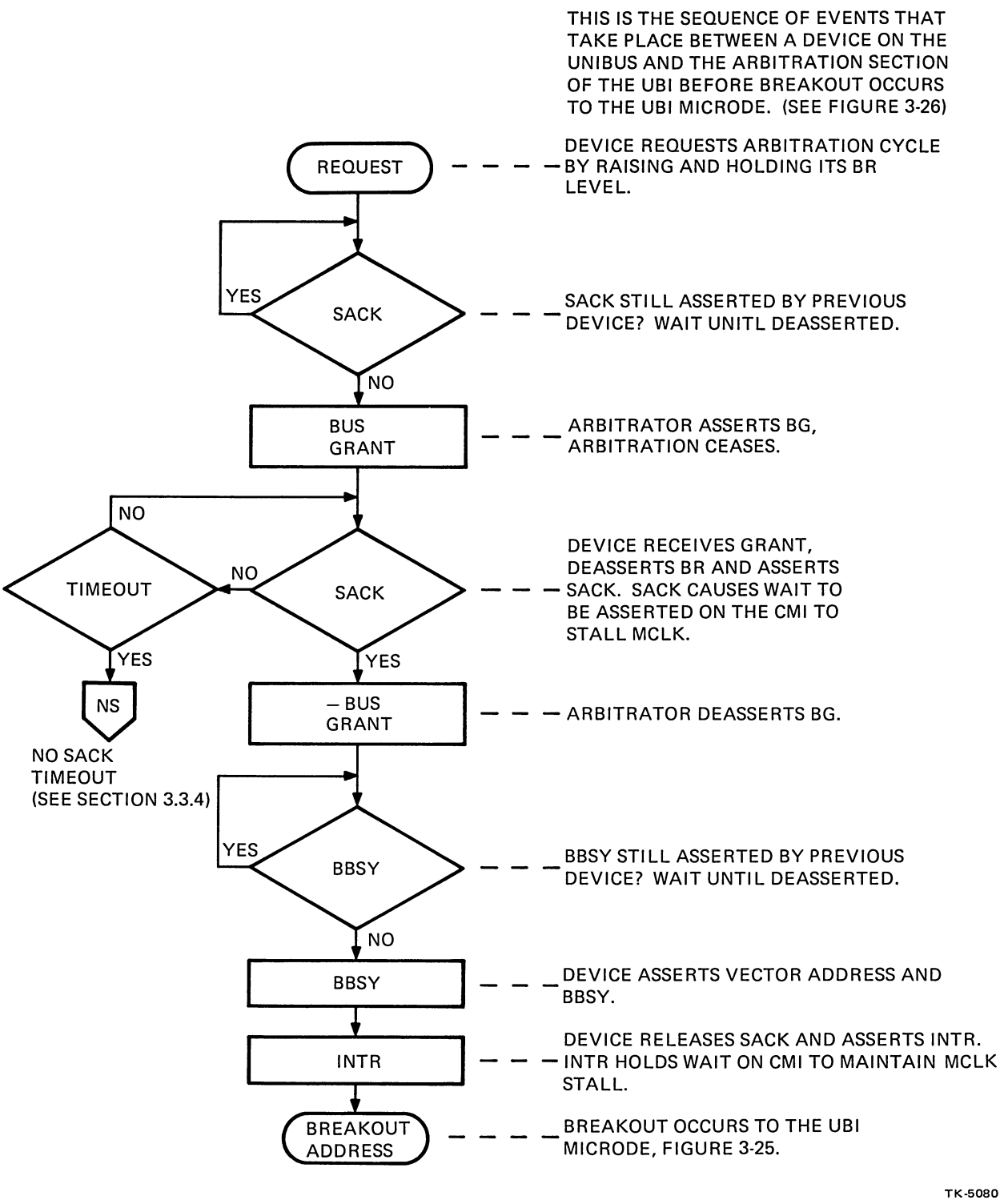


Figure 3-24 UNIBUS BR Arbitration Flow

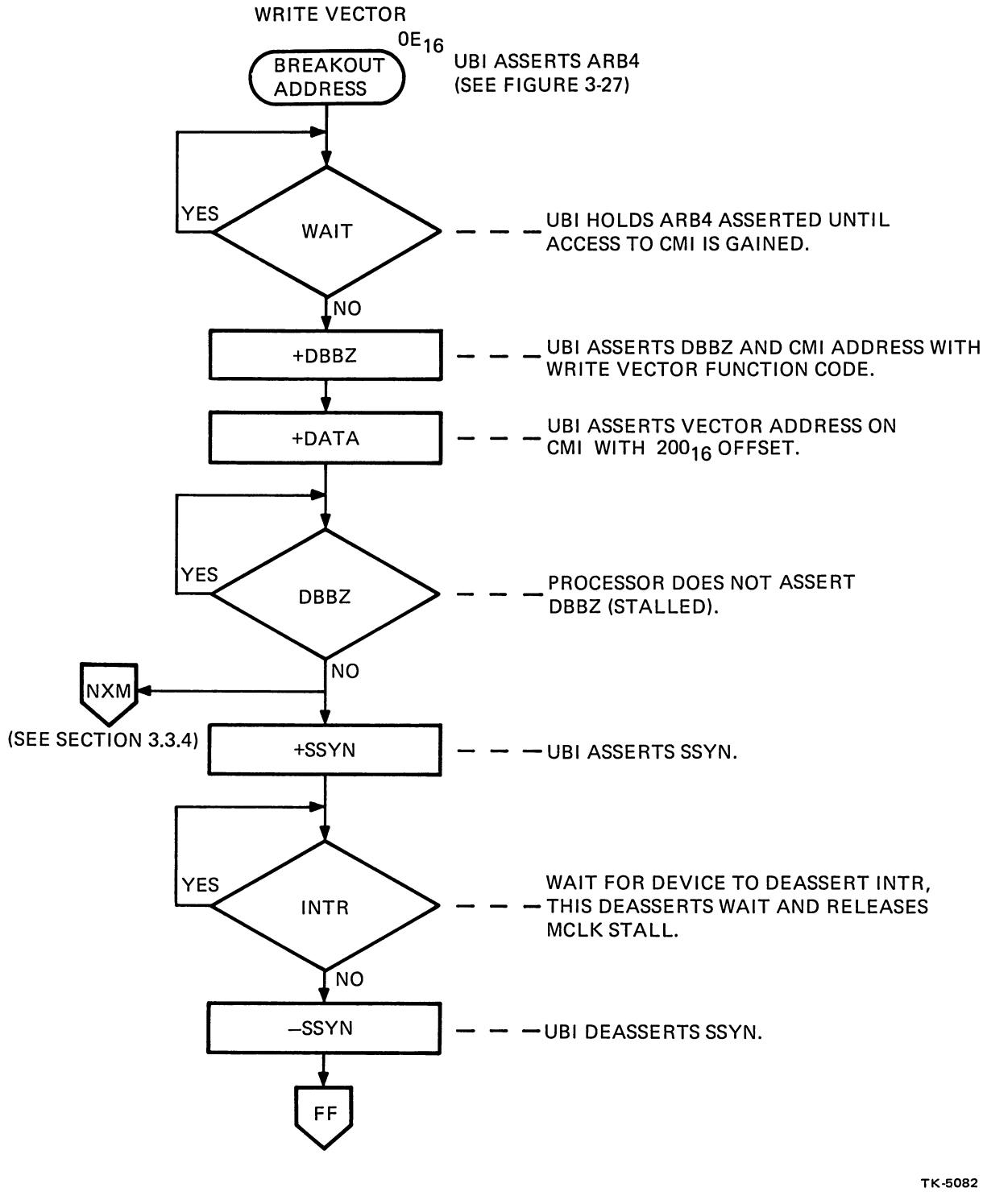


Figure 3-25 UBI Write Vector Flow Breakout Address ⟨0E⟩

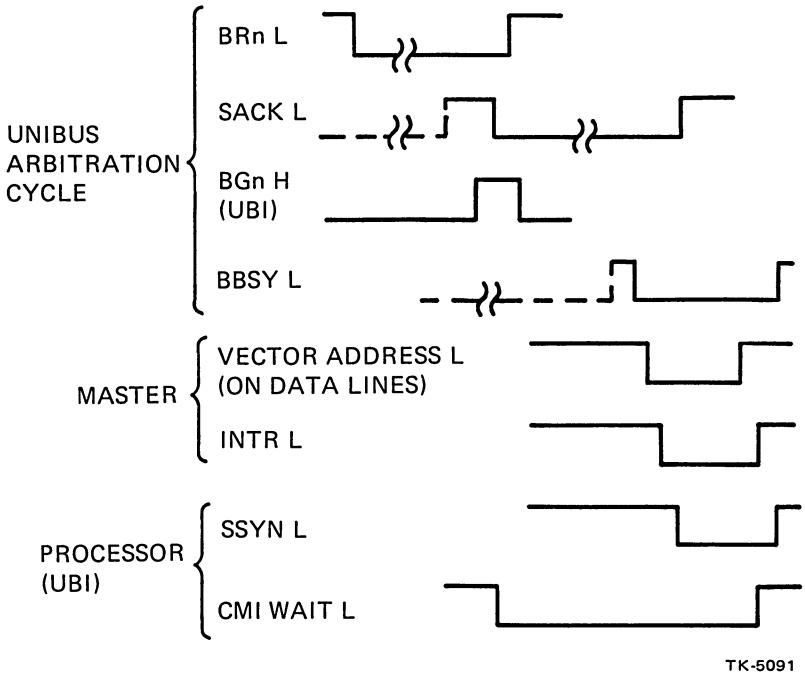


Figure 3-26 UNIBUS BR Cycle

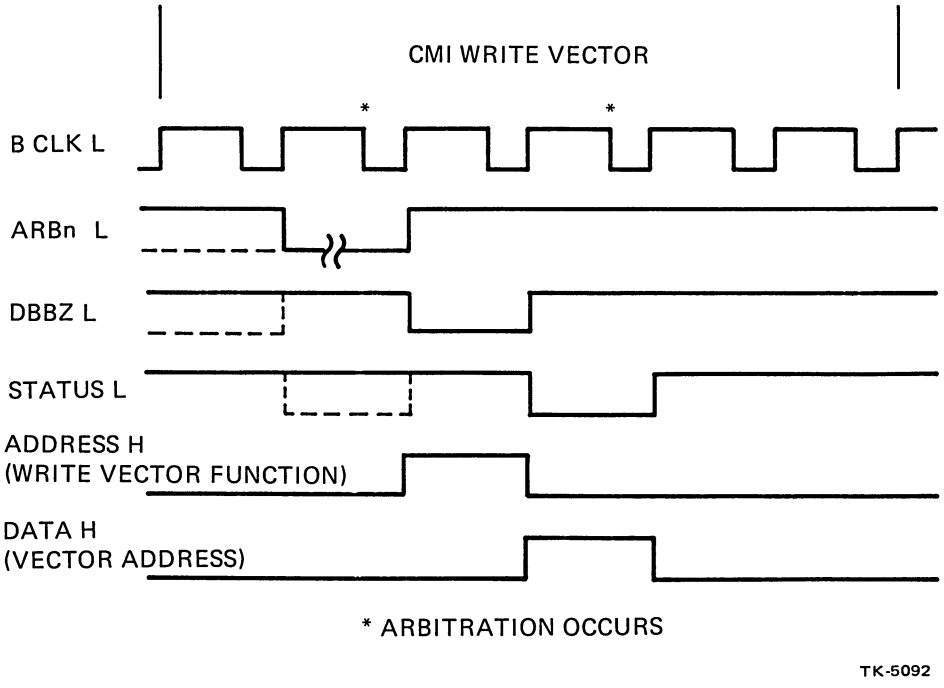


Figure 3-27 CMI Write Vector Cycle

3.3.5.1 Passive Release – The interrupt/write vector operation described above constitutes an "Active Release" of the UNIBUS device since the write vector operation was completed normally. A "Passive Release" is a condition caused by a device that raises a BR level and then, because of a malfunction or because of software or hardware limitations, loses it. If the BR level is lost after being synchronized by the arbitrator, bus grant is asserted and held to await the return of SACK. A No SACK Timeout normally causes the arbitartor to assert SACK in order to release the bus grant level.

In order to prevent a passive release from holding the processor in a stall for the duration of the SACK timeout delay, a method is provided to release the CCS from the stall. With no requesting level present while in the interrupt service microroutine, the INT chip (see Section 3.6.1) interprets the requesting level as lower than the current IPL. The bus grant enable flop is allowed to set for one bus clock cycle (fake grant), releasing the stall when it deasserts. Since a BR level is no longer asserted, no grant is issued to the UNIBUS.

3.3.5.2 BR Data Transfer – Some devices are designed to transfer data under the authority of a BR request. BR arbitartion takes place as usual with one exception: once the device asserts BBSY, it then asserts address and data as it would for an NPR, asserting MSYN instead of INTR. A UBI microsequencer breakout address is selected as for an NPR to process the transaction.

3.4 UBI MICROWORD BIT FIELD FUNCTIONS

Explanations of the UBI microword bit fields are provided for reference when use is made of the microprogram listings.

3.4.1 Single Bit Functions

Single bits of the UBI microword have the following functions:

CMI ARB — UBI priority arbitration bit (ARB4 level) is asserted during arbitration for the CMI.

SSYN — Slave sync is the microprogram response to MSYN from a master device on the UNIBUS.

MSYN — Master sync is asserted on the UNIBUS during a CPU read or write to the UNIBUS.

BUF CMI — This bit is illustrated in Figure 2-4, Address MAP, (MAP OUT EN). When set, it enables the MAP address translation to the BUF CMI lines, as shown in Figure 2-6, for assertion by the UDP as part of the CMI address.

3.4.2 UB DATA and UA CTRL Fields

The UB DATA and UA CTRL fields control the UNIBUS data and address transceivers shown in Figure 2-2, UNIBUS Interface to UBI:

UB DATA — UB data bits ⟨1:0⟩ control the tranceivers for the UNIBUS data lines,

UB DATA Bit 1	UB DATA Bit 0	Function
1	0	Receive UB Data
0	1	Transmit UB Data
0	0	OFF (Hi-Z)
1	1	Transmit UB DATA, disable PB

UB Address CTRL

UB address CTRL bits ⟨1:0⟩ control the UNIBUS address line transceivers and receiver multiplexer:

UA CTRL Bit 1	0	Function
0	0	Enable BUF CMI to UB Address lines
0	1	OFF (Hi-Z)
1	0	Receive UB address
1	1	Receive and increment UB address

3.4.3 PRTC Field

The PRTC and BDPC fields each contribute to gating and clocking of data within the UDP, illustrated in Figure 2-3, UDP Data Flow:

Port control (PRTC) field bits ⟨2:0⟩, Table 3-2, control the UB data, UB address, BUF CMI, and CMI data port drivers of the UDP, multiplexer gating to the ports, and BDP/BAR registers which are clocked by B CLK L as selected by the BDPC field.

UB address port and CMI data port drivers in each UDP chip are enabled by flip-flops set by the enabling PRTC codes.

PRTC ⟨2⟩ on a 1 sets a flop that enables the CMI mux to receive CPU read or UNIBUS DATO(B) data from the byte swap mux or BDP registers as it disables the BUF CMI receivers. A flop is set at the end of the B CLK L cycle when the PRTC code first appears. It remains set for one B CLK L cycle after the enable code disappears.

Table 3-2 PRTC Control for UDP Gating

PRTC Bits 2	1	0	Function
0	0	0	Idle, tristate drivers disabled, all ports receive data
1	1	0	Not Used
			These codes are found during a CPU read or write to the UNIBUS: CPU Read or Write – enable RCAR to BUF CMI mux, and BUF CMI port drivers to UB address transceivers (Figure 2-2). Disable CMI latch to BUF CMI mux and BDP registers (normally enabled). (CMI latch is always enabled to the RCAR which is clocked during the CPU-initiated CMI address cycle.) CMI latch is enabled to UB data byte select mux, BDP register output gating is disabled.
1	0	1	CPU Write – UB data port drivers are enabled, UB data byte select mux is clocked to UB data latch.

Table 3-2 PRTC Control for UDP Gating (Cont)

PRTC Bits 2 1 0	Function
1 1 1	CPU Read – CMI data port driver enable flip-flop is set, CMI mux flop is set for byte swap data. These codes are found during UBI arbitration for the CMI, and for UNIBUS NPR or purge data: PRTC bit ⟨1⟩ on a 1 disables UDP match gating during UBI arbitration to receive match level driven by UCN. When DBBZ is deasserted for one B CLK cycle and UBI is highest priority, match is driven low by the UCN. With match low, the UDP enables CMI data port drivers with MAP address translation from BUF CMI lines and the UCN asserts DBBZ. On the following B CLK cycle, for DATO(B) data, the CMI mux flop is set for byte swap data.
0 1 0	UNIBUS arbitration and MAP address translation.
0 1 1	Purge Arbitration – UA CTRL field turns OFF UB address transceiver (Figure 2-2). Flop is set enabling UB address port drivers for BAR driven MAP address translation (Figure 2-3).
1 0 0	UNIBUS NPR or purge has access to CMI – Assert DBBZ, set flop enabling CMI data port drivers with CMI address from BUF CMI lines. Set CMI mux flop, for DATO(B) or purge, and keep drivers enabled for CMI data longword. For purge, clear UB address port flop at end of CMI address cycle. BDP DATO(B) data on CMI (NPR or Purge) holds CMI data port drivers asserted with selected BDP register outputs and disables byte swap outputs. (Byte swap is enabled if DDP is selected, BDP register outputs are disabled.)
0 0 1	DDP DATI data from the CMI – Enable UB data port drivers, enable CMI latch to UB data byte select mux, disable BDP register output gating.

3.4.4 BDPC Field

Buffered data path control (BDPC) field bits ⟨2:0⟩ are ANDed with B CLK L to clock data to the BDP registers from the CMI latch or the byte swap mux. They also enable byte clocking to the UB data latch, from the byte select mux.

Table 3-3 illustrates byte clocking to a selected BDP register as directed by the states of UNIBUS address bits ⟨A1:A0⟩ and to the offset bit from the MAP.

Table 3-3 BDP Register Byte Clocking

Operation and BDPC⟨2:0⟩ Value	B3	B2	B1	B0	Offset	A1	A0
Default (BDPC = 0 0 0)	–	–	–	–	–	–	–
DATI*(BDPC = 0 0 1)	1	1	1	1	–	–	–
DATO (BDPC = X 0 1)	–	–	–	1	0	0	–
	–	–	1	–	–	0	–
	–	1	–	–	0	1	–
	1	–	–	–	–	1	–
	–	1	–	–	1	0	–
DATOB (BDPC = 1 1 X)	–	–	–	1	0	0	0
	–	–	1	–	0	0	1
	–	1	–	–	0	1	0
	1	–	–	–	0	1	1
	–	–	1	–	1	0	0
	–	1	–	–	1	0	1
	1	–	–	–	1	1	0
DATO(B) and Wrap (BDPC = 1 0 0 and increment UB Address)	–	–	–	1	1	1	(1)

*Four bytes of CMI data are always returned from main memory on a DATI.

Table 3-4 illustrates DATI clocking of CMI data bytes ⟨B3:B0⟩ from the byte select mux to the UB data latch, as directed by the MAP offset bit and UB address bit ⟨A1⟩ from the device.

Table 3-4 UB Data Latch Byte Clocking

BDPC Bits 2 1 0	UB Data Latch UDB1	UB Data Latch UDB0	Offset	A1	Offset and A1 Value
0 0 1	B1	B0	0	0	(0)
0 0 1/011	B3	B2	0	1	(1)
0 0 1/011	B2	B1	1	0	(2)
0 1 1	–	B3	1	1	(3)
0 0 1*	B0	–	1	1	(3)

*DATI wrap requires UBI read to next longword address.

BDPC code 0 0 1 enables DDP DATI data from the CMI latch to be clocked directly to the UB data latch. For a BDP DATI, this also clocks the first word of received CMI data to the UB data latch, as the longword is clocked to the BDP register.

BDPC code 0 1 1 for a BDP DATI enables the second word to be clocked to the UB data latch from the BDP register. In the case of the DATI wrap, another UBI read occurs to main memory and UB data latch byte 1 is clocked again before SSYN is returned to the device.

Table 3-5 lists BDPC bits that apply to additional gating within the UDP.

Table 3-5 BDPC Control for the UDP

BDPC Bits 2	1	0	Function
0	0	0	Default, no bytes are clocked.
0	0	1	Enable gating by which B CLK L clocks four bytes of DATI data to the selected BDP register from the CMI latch. It also directs these functions: ● Disable byte swap mux to BDP registers. ● Enable CMI latch to byte select mux. ● Enable UB data latch drivers; clock UB data latch bytes ⟨1:0⟩. If offset and A1 = 3, clock byte ⟨1⟩ only.
0	0	1/1X X	Clock BAR register for selected BDP.
1	X	X	Disable CMI latch outputs to BUF CMI mux and BDP registers.
X	1	1	Enable UB data latch drivers, clock UB data bytes ⟨1:0⟩.

3.4.5 NEXT and BUT Fields

The NEXT field is used as pointer to the address of the next microinstruction to be executed. This is a direct address if the BUT field code is at the default value of 0 (octal) since the BUT gating (Figure 2-13) is disabled. It may also point to the base address of a set of branch addresses. A branch address is selected as a result of tests or checks made in gating enabled by a specific branch under test (BUT) code.

Figure 3-2, UBI Microword, in Section 3.1.2, uses the power-up code as an example. The BUT code value is 0, the default code. The microprogram then goes directly to the address specified by the NEXT field, OF (hexadecimal), the first fork address. The first fork microinstruction contains 00 (hexadecimal) in the NEXT field, the base address for breakout addressing in Section 3.1.3.

Figure 2-13, UBI Control Store, shows UCR NXT bits ⟨6:4⟩ on a 0 low. These drive PROM address lines directly. UCR NXT bits ⟨3:0⟩ on a 0 allow UCR ⟨A3:A0⟩ to go high. They are held to a 1 low, however, by BUT field gating from the UCN as a result of a BUT value of 7 in the first fork microword.

Figure 3-28 illustrates UCN drivers for the UCR ⟨A3:A0⟩ bits. Enabled by the BUT values shown, the outputs remain at a 1 low until deasserted by the other leg of the gates. The gate for UCR A3 L, for example, is held low until first fork breakout to a BDP is requested by the UNIBUS. (FF UBUS H is true for a BUT value of 7 with MSYN asserted by the UNIBUS, with no NXM status or purge request pending.)

Table 3-6 lists conditions within the UCN which select breakout address (Section 3.1.3) for CPU transactions to the UNIBUS, and UNIBUS-initiated transactions to the UBI. The DP0 level selects the direct data path, and is deasserted when a buffered data path is selected. A 0 indicates the bit(s) deasserted to release the microsequencer from the 0F idle state.

Table 3-6 UNIBUS and CPU Breakout Address Select

BUT Code = 7 (First Fork)

UCR 3	UCR 2	UCR 1	UCR 0	Breakout Conditions
0	X	X	X	FF UBUS and $\overline{\text{AUTO PURGE}}$ and $\overline{\text{DP0}}$
X	0	X	X	FF UBUS and $\overline{\text{DATOB WRAP}}$ and DP0, or FF UBUS and C1 and $\overline{\text{AUTO PURGE}}$ and $\overline{\text{DATOB WRAP}}$
X	X	0	X	MATCH and $\overline{\text{C1}}$ and $\overline{\text{DP0}}$ and CD Flag, or MATCH and C1 and $\overline{\text{DP0}}$ and FULL and $\overline{\text{WRAP}}$, or PURGE (Purge Request or Auto Purge)
X	X	X	0	FF UBUS and C1 and DP0, or FF UBUS and CD Flag and WRAP and $\overline{\text{C1}}$ and $\overline{\text{DP0}}$, or FF UBUS and C0 and C1 and $\overline{\text{DP0}}$, or INTR and $\overline{\text{PURGE}}$
X	0	0	X	FIRST FORK and LATCH ADDU* (CPU Read on UNIBUS)
X	0	0	0	FIRST FORK and LATCH ADDU* and BUF CMI 27 (CPU Write to the UNIBUS)

0 = Deasserted to zero (high).

*Latch ADDU flop is set in the UCN by CPU access to UNIBUS address space. See Section 2.5.3.4 and Section 2.6.1.

When breakout occurs to the microprogram, UCR A3 is not selected by the UCN but is driven from the NEXT field. UCR ⟨A2:A0⟩ then select up to eight possible branch addresses unless a bit(s) is driven to a 1 by the NEXT field. Table 3-7 lists conditions tested by the BUT field during microprogram execution.

3.5 CMI ACCESS TO UBI

3.5.1 Slave Control (SC)

Figures 3-29 and 3-30 illustrate those functions of the UCN slave control logic that allow CPU access to the MAP and control status registers.

Table 3-7 BUT Code Tests

BUT Code Value	Tests For
7	Return to first fork
6	DATO/DATI wrap, or DBBZ or NXM from CMI (DATOB wrap is tested for on breakout to the microcode)
5/4	MSYN or SSYN from UNIBUS, or timeout (5 = clock the byte flags or CD flag)
3/2	MSYN from UNIBUS, or WON CMI (WON CMI = ARB4 asserted, UBI is highest priority, and DBBZ is not asserted on CMI) [3 = clock 0 0 0 1 to byte flags on DATO(B) wrap]
1	MSYN from UNIBUS, or empty purge (no byte flags set on purge)
0	Default, no BUT gating enabled

The slave control consists of a stepping register of three bits within the UCN labeled SST2, SST1, and SST0. The SC1 output from the UCN is driven from the zero side of SST1, the SC0 output from the one side of SST0. In the idle state with all flops cleared, the SC1 H output is high, the SC0 H output is low.

Figure 3-29 shows slave control response to a read generated by the CPU (see Section 2 6.1.1). SC ⟨1:0⟩ outputs, when not in the idle state, enable the UB address port drivers with the contents of the RCAR. Access gating is enabled as described in Section 2.5.3.3. SST2 set asserts the DBBZ level onto the CMI and SC0 H high enables the CMI data port drivers from the UDP with MAP or CSR contents received on the BUF CMI lines (see Figure 2-3). SST1 on a 1 selects status ⟨1:0⟩ bits to a 1 to return no error status to the CPU. In Figure 3-29:

SC ⟨1:0⟩ ≠ Idle	Enable UB address port drivers with RCAR contents.
UBI UB Address 08 = 1	Enable MAP output drivers to BUF CMI lines. (See Figure 2-10)
UBI UB Address 08 = 0	Enable CSR outputs from UCN to BUF CMI lines.
SC0 H = H	Enable CMI data port drivers with received BUF CMI data.
SST1 H = H	Enable STATUS ⟨1:0⟩ = No Error

Figure 3-30 shows the slave control sequence for a CPU write to a UBI address. SC1 H and SC0 H both low enable data received on the CMI data lines to the BUF CMI drivers; and SST 1 on a 1 returns no error status to the CPU. DBBZ is not asserted in this case since the write data is clocked in one B CLK cycle by the UBI. In Figure 3-30:

SC ⟨1:0⟩ ≠ Idle	Enable UB address port drivers with RCAR contents.

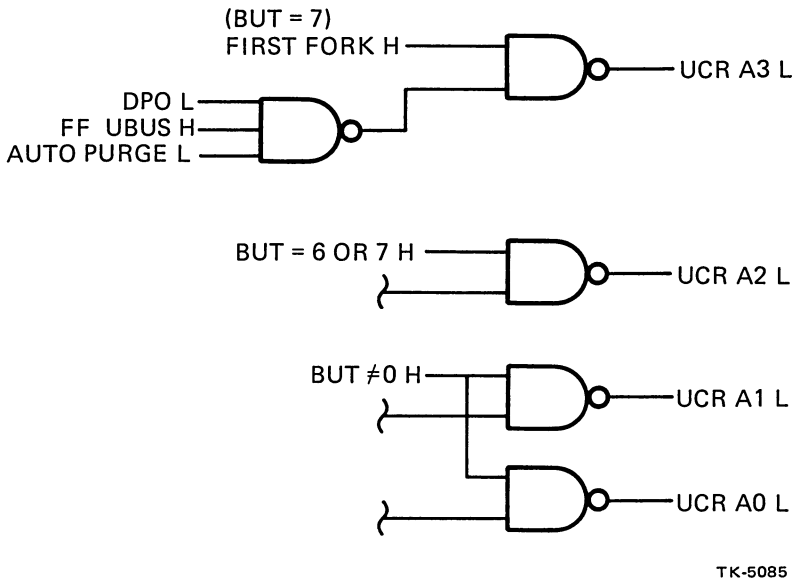


Figure 3-28 UCN BUT Field Gating

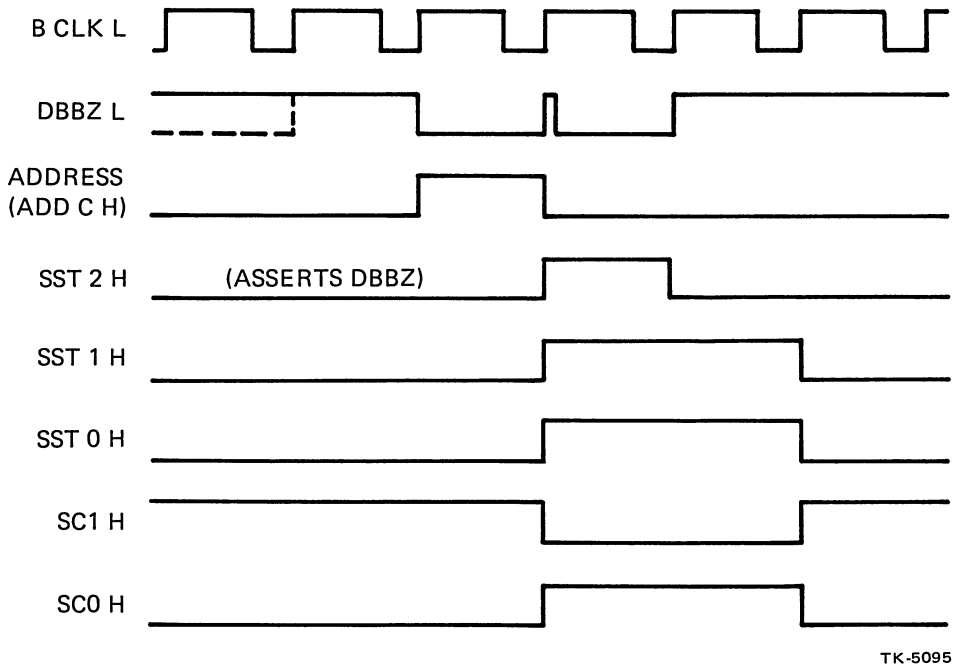


Figure 3-29 CPU Read From UBI Cycle

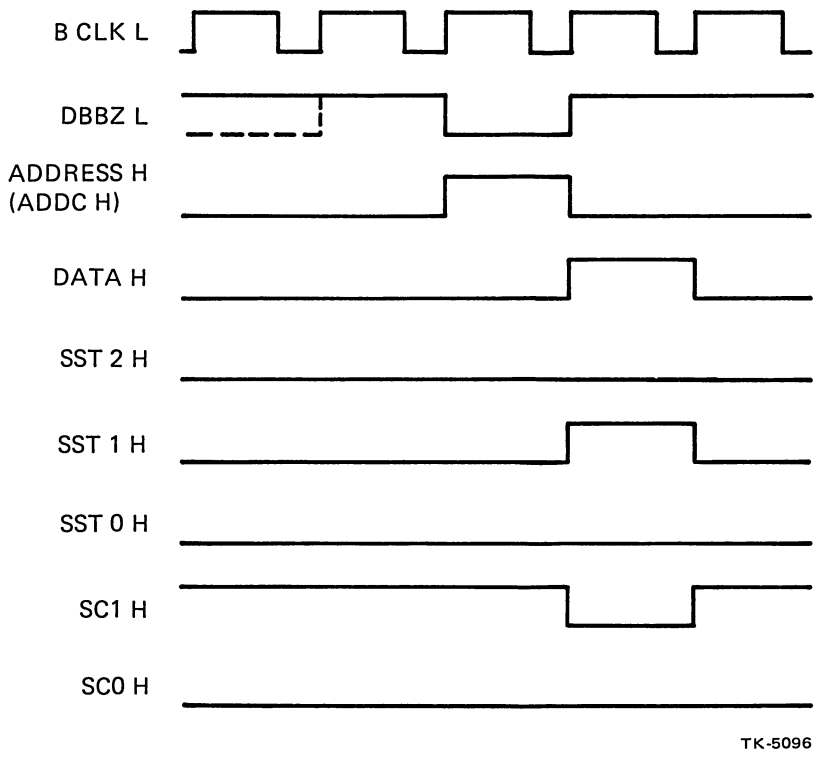


Figure 3-30 CPU Write to UBI Cycle

SC1 H = L	Enable BUF CMI drivers to MAP and CSR inputs.
SC0 H = L	UBI UA08 = 1, clock WRITE MAP signal. UBI UA08 = 0, clock CSR bits ⟨30,29,00⟩ (See Figure 2-10)
SST1 H = H	Enable STATUS ⟨1:0⟩ = No Error

3.6 PROCESSOR LOGIC

This is a summary of logic on the UBI which is part of the processor.

3.6.1 System Interrupts (INT)

The arbitration section of the CPU control store (CCS) consists of the INT chip with external TTL circuitry. Interrupts are generated for conditions which steer the CCS via the microvector lines.

- System Power Fail
- Write Bus (WBUS) Error
- Corrected Memory Data
- Interval Timer
- Console Device (LA34)
- UNIBUS Interrupts

The WCTRL field of the CCS commands the INT chip to read or write status register data from the WBUS, issue UNIBUS grant, or to place status data or REI check results onto the microvector lines. These functions:

1. Save and return on the WBUS, the AST level; and the IS, CURMODE, PRVMODE, and IPL values of the PSL.

2. Receive and store the HSIPR (Highest Pending Software Interrupt Priority Request) used in interrupt arbitration.

3. Perform REI check calculations.

The INT chip receives the processor initialize signal which clears all registers. The uVCTR branch signal indicates that the microvector lines are being read to process the highest pending priority interrupt, and the corresponding interrupt latch can be cleared.

The PTE CHK OR PROBE and the UTRAP levels indicate to the INT that service on one of these operations is in progress. DOSRV indicates the presence of hardware service conditions and requests the UTRAP sequence.

The microvector interrupt code (UVIC) transmitted on MICROVECTOR ⟨2:0⟩ lines indicates to the processor the interrupt priority level (IPL) of a requesting device.

IPL	Requesting Device	UVIC
1E	SPFI – Power Fail	111
1D	WEI – Write Error	110
18	CDI – Corrected Memory Data	100
17	TIMER INT – Interval Timer	011
⟨17:14⟩	SBR ⟨7:4⟩ – UNIBUS	010
14	SLINE INT – Console	001
⟨0F:01⟩	HSIPR – Software	000
00	(no request)	000

3.6.2 Console Interface (CON)

Two CON chips provide asynchronous serial line interfacing between the CPU and the console terminal and TU58. The CON chip contains limited character recognition of received characters from the console and can request both micro and macro level interrupts.

Communication between the CPU and the CON chip takes place by received commands on the WCTRL bus and transmit/receive data on the WBUS. Access to internal registers is gained by first loading WBUS bits ⟨23:22⟩ to the console register address register (CRAR) or to the tape register address register (TRAR).

Table 3-8 lists CRAR/TRAR codes that enable read/write WBUS transfers to the following registers:

1. Transmitter data buffer of 8 bits (CTDB/TTDB)

2. Transmitter control/status register of 2 bits (CTCSR/TTCSR)

3. Receiver data buffer of 8 bits (CRDB/TRDB)

4. Receiver control/status register of 2 bits (CRCSR/TRCSR)

5. Console status register (CSR) is not applicable to the tape control and consists of 2 bits, HALT and HALT PENDING.

Table 3-8 CRAR/TRAR Code

Register	CRAR/TRAR Register Address	Read/Write
CTDB/TTDB	00	W
CTCSR/TTCSR	01	R/W
CRDB/TRDB	00	R
CRCSR/TRCSR	10	R/W
CSR/ –	11	R/W

The baud rate for the TU58 is fixed at 19,200 baud. The baud rate for the console terminal is set by grounding the backplane pins listed in Table 3-9 on the RDM module, slot 6-C.

Table 3-9 Console Baud Rate Select

CON BR D Pin C50	CON BR C Pin C49	CON BR B Pin C46	CON BR A Pin C45	Baud Rate
0	1	0	0	300
0	1	0	1	400
0	1	1	0	600
0	1	1	1	1200
1	0	0	0	2400
1	0	0	1	3600
1	0	1	0	4000
1	0	1	1	4800
1	1	0	0	7200
1	1	0	1	9600
1	1	1	0	19200
1	1	1	1	38400

0 = Open
1 = Ground

3.6.3 Time of Year (TOY) Clock

The TOY clock consists of a 32-bit 1-Khz counter with a 16 × 4-bit RAM as offset memory. WCTRL code bits ⟨5:0⟩ are defined as follows:

Load Offset Memory, Clear TOY counter = 001101

Read TOY clock = 001001

WBUS bits are transmitted for following control functions:

WBUS Bit	Function
⟨22⟩ = 1	Reset TOY Counter
⟨21⟩ = 1	Enable Offset Memory Outputs
⟨20⟩ = 1	Disable Write to Offset Memory Inputs
⟨19:18⟩	Offset Value written for selected byte

⟨16⟩	Counter Select 1 (Read/Write)
⟨17⟩	Counter Select 0 (Read/Write)

Obtaining the current value of TOY clock contents consists of four consecutive reads with counter select ⟨1:0⟩ values from 0 through 3. Bytes 3 through 0 (32 bits plus offset) are read to the WBUS as follows:

WBUS Bit	Definition
⟨27:26⟩	Offset Value
⟨25:24⟩	Byte Identity ⟨3:0⟩
⟨23:16⟩	Eight bits of Selected Byte

3.6.4 SID System Revision Level

WCTRL ⟨5:0⟩ code 010001 reads the hardware revision level to WBUS ⟨23:16⟩ to construct the SID longword. Pins which are grounded to select a 1 output to the WBUS are identified on UBI slot 4 as listed in Table 3-10. Ungrounded pins should be connected to +5 V through a pullup resister.

Table 3-10 SID System Revision Level

Signal Name	UBI Pin on Slot #04, Connector B
+5 Volts	B38
Ground	B43 and B44
SYS ID 7	B46
SYS ID 6	B48
SYS ID 5	B49
SYS ID 4	B50
Ground	B51 and B52
SYS ID 3	B53
SYS ID 2	B54
SYS ID 1	B55
SYS ID 0	B56
+5 Volts	B58

APPENDIX A
UNIBUS Exerciser/Terminator (UET)

The M9313 module is located in the output slot of the last device on the UNIBUS, terminating the end of the UNIBUS. It provides for diagnostic testing of the UBI's basic capabilities to handle UNIBUS addressing, data transfers, and interrupts.

A.1 UET REGISTER VECTOR ADDRESSES

772140 = Address Register ⟨A15:A00⟩ (Word load only to this register, byte loading causes timeout.)

772142 = Data Register ⟨D15:D00⟩

772144 = Control Register, CR ⟨15:00⟩

772146 = PROM Read-only Register ⟨D07:D00⟩

A.2 CONTROL REGISTER (CR) BITS

CR⟨0⟩ = *NPR, Write 1 to initiate transfer.

CR⟨2,1⟩ = ⟨C1,C0⟩ transfer command bits;
0 0 = UET DATI
0 1 = UET DATIP
1 0 = UET DATO
1 1 = UET DATOB

CR⟨4,3⟩ = ⟨A17,A16⟩ high-order UNIBUS addressing bits (not cleared by UET INIT).

CR⟨5⟩ = PB, Parity bit, simulates memory Parity Error for UET DATI

CR⟨6⟩ = TO, Timeout, SSYN not returned. Clocked on each transfer and cleared by UET INIT.

CR⟨7⟩ = PE, Parity Error, detected during UET DATI. Clocked on each UET DATI and cleared by UET INIT.

CR⟨11:08⟩ = *⟨BR7:BR4⟩, Write 1 to initiate interrupt.

CR⟨14:12⟩ = Not used.

CR⟨15⟩ = UET INIT, Initialize UET to simulate Reset or Power-Up. Write-only bit, always reads as 0. Does not clear CR ⟨4,3⟩.

*CR⟨11:08,00⟩ (⟨BR7:BR4,NPR⟩) remain set until cleared by writing a 0 or by UET INIT. Multiple interrupts occur if more than one bit is set.

A.3 NPR DATA TRANSFERS

UET Write:
Load Address Register ⟨A15:A00⟩
Load Data Register ⟨D15:D00⟩
Load Control Register to initiate transfer;
CR⟨0⟩, = Generate NPR
CR⟨2,1⟩ = 10 for DATO, 11 for DATOB
CR⟨4,3⟩ = ⟨A17,A16⟩ of UNIBUS Address

UET Read:
Load Address Register ⟨A15:A00⟩
Load Control Register to initiate transfer;
CR⟨0⟩, = Generate NPR
CR⟨2,1⟩ = 00 for DATI, 01 for DATIP
CR⟨4,3⟩ = ⟨A17,A16⟩ of UNIBUS Address

A.4 BR INTERRUPTS

Load Data Register ⟨D15:D00⟩ = Vector Address
Load Control Register CR⟨11:08⟩ = BR⟨7:4⟩ level

A.5 PROM TRANSFERS

Load Address Register ⟨A11:A00⟩ = PROM Data Address (4K)
Read PROM Data Register ⟨D07:D00⟩ = PROM Data (8 bits)
PROM Data Register ⟨D15:D08⟩ = Undefined

VAX-11/750 UNIBUS Interface
Technical Description
EK-UI750-TD-001

Reader's Comments

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc.? Is it easy to use? _____

_____

_____

_____

What features are most useful? _____

_____

_____

_____

What faults or errors have you found in the manual? _____

_____

_____

_____

Does this manual satisfy the need you think it was intended to satisfy? _____

Does it satisfy *your* needs? _____ Why? _____

_____

_____

_____

☐ Please send me the current copy of the *Technical Documentation Catalog*, which contains information on the remainder of DIGITAL's technical documentation.

Name _____ Street _____
Title _____ City _____
Company _____ State/Country _____
Department _____ Zip _____

Additional copies of this document are available from:

Digital Equipment Corporation
444 Whitney Street
Northboro, MA 01532
Attention: Printing and Circulation Services (NR2/M15)
Customer Services Section

Order No. EK-UI750-TD

Fold Here

Do Not Tear — Fold Here and Staple

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD, MA.

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation
Communications Development and Publishing
1925 Andover Street
Tewksbury, Massachusetts 01876

BLANK

BLANK

BLANK